

WebFOCUS

WebFOCUS Creating Financial Reports
Version 5 Release 2

EDA, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPpack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks, and iWay and iWay Software are trademarks of Information Builders, Inc.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2003, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

Preface

This documentation describes how to use the Financial Report Painter. It is intended for application developers and others who are responsible for creating financial reports.

How This Manual Is Organized

This manual includes the following chapters:

Chapter/Appendix		Contents
1	Financial Reporting and Analysis for the Enterprise	<p>Introduces the WebFOCUS Financial Reporting Platform— Information Builder’s single vendor enterprise business intelligence solution — and describes the role of of the WebFOCUS Financial Report Painter at the heart of that solution.</p> <p>The Financial Report Painter is designed for the special needs associated with creating, calculating, and presenting financially oriented data such as balance sheets, consolidations, profit and loss statements, budgets, and certain government mandated financial reports.</p>
2	Creating Reports With Financial Report Painter	<p>Describes how to use the WebFOCUS Financial Report Painter to model a range of specialized financial reports. These reports are distinguished from other WebFOCUS reports because calculations are inter-row as well as inter-column.</p> <p>Using an intuitive matrix design, the Financial Report Painter enables you to generate real-time reports that handle large volumes of data, volatile data relationships, and custom charts of accounts.</p>
A	Creating Financial Reports With FML Language	<p>Describes all aspects of the Financial Modeling Language (FML). This information is intended as a supplement to the documentation on the Financial Report Painter for users who wish to examine the underlying code.</p>
B	Describing Data for an FML Hierarchy	<p>Documents the syntax required in a Master File to support FML hierarchies. It provides the syntax for manual coding as well as the corresponding entries in the graphical Master File Editor.</p>

Documentation Conventions

The following conventions apply throughout this manual:

Convention	Description
<code>THIS TYPEFACE</code> or <code>this typeface</code>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable) in a text paragraph, a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option you can click or select.
this typeface	Highlights a file name or command in a text paragraph that must be lowercase.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices; type one of them, not the braces.
[]	Indicates a group of optional parameters. None are required, but you may select one of them. Type only the parameter in the brackets, not the brackets.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (...).
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Creating Reports With Graphical Tools

Creating Reports With WebFOCUS Language

To view a current listing of our publications and to place an order, visit our World Wide Web site, <http://www.informationbuilders.com>. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about WebFOCUS Developer Studio?

Call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your Developer Studio questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Information You Should Have

To help our consultants answer your questions most effectively, please provide the following information when you call:

- Your six-digit site code (xxxx.xx).
- Your Developer configuration:
 - The front-end you are using, including vendor and release.
 - The communications protocol (for example, TCP/IP or HLLAPI), including vendor and release.
 - The software release.
 - The server you are accessing, including release (for example, 5.2.1).
- The stored procedure (preferably with line numbers) or FOCUS commands being used in server access.
- The name of the Master File and Access File.
- The exact nature of the problem:
 - Are the results or the format incorrect? Are the text or calculations missing or misplaced?
 - The error message and return code, if applicable.
 - Is this related to any other problem?
- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating system are you using? Has it, WebFOCUS, your security system, communications protocol, or front-end software changed?
- Is this problem reproducible? If so, how?
- Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two data sources, have you tried executing a query containing the code to access a single data source?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes your opinions regarding this manual. Please use the Reader Comments form at the end of this manual to relay suggestions for improving the publication or to alert us to corrections. You can also use the Documentation Feedback form on our Web site, <http://www.informationbuilders.com>.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site (<http://www.informationbuilders.com>) or call (800) 969-INFO to speak to an Education Representative.

Contents

1. Financial Reporting and Analysis for the Enterprise	1-1
Enterprise Financial Reporting: A Critical Cornerstone of Business	1-2
Financial Reporting Within and Beyond an Organization	1-3
Internal Financial Reports	1-3
External Financial Reports	1-3
Financial Reporting and Government Regulations	1-4
The WebFOCUS Financial Reporting Platform	1-5
Data Consolidation/Aggregation	1-5
Report Generation	1-5
Financial Analysis	1-6
Distribution	1-7
Archiving	1-8
WebFOCUS Financial Report Painter	1-8
WebFOCUS Financial Report Painter's Advantage Over SQL-based Financial Reporting Tools ..	1-10
WebFOCUS Financial Report Painter's Advantage for Data Access	1-11
Learning More About the WebFOCUS Financial Report Painter	1-11
2. Creating Reports With Financial Report Painter	2-1
Starting a Financial Report	2-2
Designing Your Financial Report	2-4
Viewing Source Code	2-9
Previewing Your Report	2-10
Running a Financial Report	2-12
Retrieving FOR Field Values From a Data Source	2-13
Reporting Dynamically From a Hierarchy	2-21
Displaying an FML Hierarchy	2-26
Consolidating an FML Hierarchy	2-31
Supplying Data Directly	2-45
Performing Inter-Row Calculations	2-47
Inserting Rows of Text	2-57
Suppressing the Display of Rows	2-59
Saving and Retrieving Intermediate Report Results	2-60
Posting Data	2-60
Formatting Financial Reports	2-64
Adding Underlines	2-64
Formatting Columns, Rows, and Cells	2-66
Drilling Down From a Column, Cell, or Row	2-83
Adding, Inserting, and Deleting Rows	2-85
Editing Row Types and Properties	2-86
Adding and Deleting Columns	2-87

A.	Creating Financial Reports With FML Language	A-1
	Reporting With FML	A-2
	Creating Rows From Data	A-5
	Creating Rows From Multiple Records	A-8
	Using the BY Phrase in FML Requests	A-15
	Supplying Data Directly	A-16
	Performing Inter-Row Calculations	A-18
	Referring to Rows in Calculations	A-19
	Referring to Columns in Calculations	A-22
	Referring to Column Numbers in Calculations	A-23
	Referring to Contiguous Columns in Calculations	A-24
	Referring to Column Addresses in Calculations	A-25
	Referring to Relative Column Addresses in Calculations	A-26
	Referring to Column Values in Calculations	A-27
	Referring to Cells in Calculations	A-28
	Using Functions in RECAP Calculations	A-30
	Inserting Rows of Free Text	A-32
	Adding a Column to an FML Report	A-35
	Creating a Recursive Model	A-37
	Reporting Dynamically From a Hierarchy	A-39
	Requirements for FML Hierarchies	A-40
	Displaying an FML Hierarchy	A-42
	Consolidating an FML Hierarchy	A-46
	Loading a Hierarchy Manually	A-54
	Customizing a Row Title	A-57
	Formatting an FML Report	A-59
	Suppressing the Display of Rows	A-77
	Suppressing Rows With No Data	A-78
	Saving and Retrieving Intermediate Report Results	A-79
	Posting Data	A-79
	Creating HOLD Files From FML Reports	A-82
B.	Describing Data for an FML Hierarchy	B-1
	Data Requirements for FML Hierarchies	B-2
	Coding an FML Hierarchy in a Text Editor	B-2
	Defining an FML Hierarchy in the Master File Editor	B-5

CHAPTER 1

Financial Reporting and Analysis for the Enterprise

Topics:

- Enterprise Financial Reporting: A Critical Cornerstone of Business
- Financial Reporting Within and Beyond an Organization
- Financial Reporting and Government Regulations
- The WebFOCUS Financial Reporting Platform
- WebFOCUS Financial Report Painter
- WebFOCUS Financial Report Painter's Advantage Over SQL-based Financial Reporting Tools
- WebFOCUS Financial Report Painter's Advantage for Data Access
- Learning More About the WebFOCUS Financial Report Painter

WebFOCUS provides a total enterprise financial reporting solution that significantly reduces development time and maintenance costs while providing flexibility, accessibility, and accuracy in enterprise-wide financial activities.

WebFOCUS real-time Web-based financial reporting delivers accurate and timely knowledge of current and future operations, performance, and financial status for organizations' mission-critical financial needs, including:

- Budgeting
- Planning
- Consolidation
- Forecasting

From accountants to financial analysts to IT personnel, whether users are technical or nontechnical, the entire organization will benefit. CFOs will find WebFOCUS financial reporting especially useful for generating balance sheets, and CEOs will appreciate its ability to focus the organization on growth and cash flow to help fund a continual stream of product innovations, acquisitions and to meet government regulations.

Integral to it all, the powerful WebFOCUS Financial Report Painter enables organizations to build these specialized financial reports that are critical to a wide range of departments, knowledge workers, managers, suppliers, lenders, investors, and government agencies.

Enterprise Financial Reporting: A Critical Cornerstone of Business

Financial information is a critical cornerstone of business. Nearly every organization needs to be able to access financial data and turn it into useful financial information to understand where their business has been, know where it is now, and plan for where it will be in the future.

To achieve this, organizations must have effective technology and financial tools to track income, expenses, assets, and liabilities in an organized manner. They must have ways to convert data to information that can provide frequent, consistent, and accurate insight into financial operations and performance. They must have financial information that enables them to look forward in order to plan and forecast for the future. Just as important, they must have financial information that enables them to understand what went wrong in the past without wasting valuable time and resources validating, justifying and auditing previous financial situations.

Efficient and effective financial reporting solutions depend on cost effectiveness, usefulness, and flexibility. To be cost effective, the benefits of information must outweigh the cost of providing it. To be useful, information must be understandable, reliable, relevant, accurate, and timely. Such financial reporting solutions need to accommodate a wide range of users and be able to efficiently and flexibly handle changing information requirements. In addition, and no less important, financial reporting solutions must be able to provide reliable data consolidation and aggregation, easy report generation, distribution and archiving, and financial analysis. Since these demands are rigorous, few financial reporting solutions can meet the requirements.

Today, mergers, acquisitions, global expansion, investor concerns, and government regulations have made the need for accurate and real-time delivery of financial information that meets these requirements even more critical for business operations.

From accounting to decision making, WebFOCUS financial reporting provides organizations with the solutions that meet their varied needs for obtaining, evaluating, and delivering a total view of their financial status, including:

- Greater availability of financial information.
- More timely reporting processes.
- More detailed financial analysis.
- More individuals interacting with financial data.
- Increased responsibility of sharing and management of financial information.

Financial Reporting Within and Beyond an Organization

Financial reports impact business in myriad ways, especially organizations' financial and managerial accounting needs. Organizations may need financial reports to describe their financial condition and provide a framework for examining past performance, current financial situations, possible future performance, and the financial positions and performances of various employees, groups, and departments within and outside of these organizations. Organizations may require reports that describe their financial position at a particular point in time and their financial performance during a specified range of time.

Internal Financial Reports

Internally, financial reports may be required by various levels of managers within the organization to evaluate and make decisions about current and future operations. The reports may require highly detailed or summarized financial information, depending on management requirements for decision making. These internal financial reports may include sales reports, production cost reports, or other detailed financial reports.

External Financial Reports

Externally, financial reports may be required by people or agencies outside the organization who have an interest in the organization's business activities. For example, suppliers, lenders, investors, and government agencies may need to access the financial status of a business for any number of reasons. External financial reports may include balance sheets, income statements or cash flow statements.

Financial Reporting and Government Regulations

While financial reporting is critical to the inner workings of every business, it is becoming even more critical to the outside stakeholders of business. Evidence of this can be seen in the demands of investors and government for financial reporting reforms in the wake of recent corporate accounting fiascoes and financial improprieties. Thus, the demand for better, more timely financial information tailored for a wider audience is more urgent today than ever, and by all indications the trend will continue.

For example, the Sarbanes-Oxley Act contains key provisions to prevent corporate fraud and restore investor confidence in financial reports, highlighting the need for organizations to produce better financial information and be more accountable. To meet the demands of government agencies, investors, creditors, and others for accurate and current financial information, organizations must be able to disclose correct and timely financial information in varied report styles and formats for distribution not only within the organization, but outside of it. While financial manipulation resides in the province of ethics, which no financial reporting solution can change, financial reporting solutions must be effective to help eliminate misleading financial statements and valuations. Organizations must put in place financial reporting solutions that, for example, promptly recognize reversals of income that had been booked on the basis of projections which later may be proven to have been inaccurate. Financial reporting solutions must be able to generate objective and accurate valuations of assets and clear provable balance sheets in a timely manner.

The WebFOCUS Financial Reporting Platform

The WebFOCUS Financial Reporting Platform is an enterprise business intelligence solution that can access, develop, forecast, archive, and report on vital financial information from a wide range of data sources, across many departments and business units.

Data Consolidation/Aggregation

Consolidating data based on Charts of Accounts or other hierarchies is a powerful feature that enables real-time data access for timely and accurate financial reports. Typically, data is stored at the posted account level in the database, however, if users want to report on assets or liabilities they first must consolidate the lower level accounts to higher levels. For example, if users want to print a report, such as an income statement, they need the numbers for Total Revenue. Total Revenue is not typically stored in the database. However, the revenue numbers are all posted in the database at low levels. Thus, some sort of mechanism is required so that at report time these lower level numbers show up as consolidated numbers, or numbers rolled up to a higher level. These consolidated (rolled up) numbers can then be shown as Total Revenue in the financial statement.

WebFOCUS consolidation of financial data provides the benefits of quickly and easily managing financial reporting without the overhead involved in cube creation. It also speeds the creation of data warehouses to make it easier to generate financial reports with more flexibility when dealing with various business mergers, acquisitions, and new business units.

Report Generation

WebFOCUS report generation makes it easy to build a wide range of financial reports for many uses and many audiences. Organizations can get a worldwide picture of financial and business activities because WebFOCUS financial reports can handle international currency conversion, reading and interpreting national characters, and handle sorting, case conversions and formatting of dates, currencies, and numbers. Multiple currencies can be shown on a single report with appropriate symbols, and reporting applications can be developed that provide users with the ability to select the language in which they are most comfortable working.

Additionally, its dynamic posting capability allows for generating summary statements from individual, detailed reports. And its ability to dynamically roll-up financial data that has been consolidated simplifies fast generation of a range of financial reports. Refer to *WebFOCUS Financial Report Painter* on page 1-8 for more information.

Financial Analysis

WebFOCUS financial reports that contain active drill-downs can take advantage of WebFOCUS OLAP. Within a financial report, for example, a user can drill down on a field such as Income Statement. The drill-down takes the user to a non-financial report, from which the user can access a WebFOCUS OLAP panel. Once in the WebFOCUS OLAP panel, the user can perform extensive data analysis functionality on the organization. For example, the user can slice and dice data in many ways to answer vital business questions such as "What's selling and at what price point?" or "What products in what geographical location account for the largest percentage of my profits?"

WebFOCUS financial reports, with drill-downs intact, can be automatically transferred to and displayed in Excel. This conversion greatly reduces the possibility for errors. Report elements, including field names, headers, subtotals, sort breaks, and drill-downs are preserved within Excel. Clicking on embedded hyperlinks in Excel produces the same detailed reports as those created in a WebFOCUS financial report. With report information transposed into Excel, users can take advantage of pivot tables and other features. In addition, WebFOCUS supports a direct save of report data to Pivot Tables, which provide a powerful data mining capability. By starting with a relational request to get the scope of desired data into a single report, and then automatically generating a Pivot Table for iterative exploration, this feature supports a workflow common to many organizations. Automatic generation of Pivot Tables is a WebFOCUS exclusive.

WebFOCUS financial reports can also take advantage of Knowledge Mapping, a non-linear problem solving process. Knowledge Mapping lets users investigate internal and external financial information sources, pass valuable information to other financial reports or graphs, and map into applications such as spreadsheets.

WebFOCUS forecasting features can help executives identify trends in numeric data and predict values beyond the range of values stored in the data source, enabling business management to anticipate future conditions and plan ahead.

Distribution

Once financial reports are generated, organizations can use their existing technology and data-communication infrastructures to deliver reports across the enterprise in a number of ways, such as e-mail, the Web, and wireless devices. Financial reports can be distributed:

- On a scheduled or event-driven basis.
- Via automatic bursting, delivering reports to each user based on his or her security profile, access permissions and so forth. This more efficiently makes use of network resources.
- By setting up portal-like interfaces from which users can access financial reports.
- By incorporating financial reports into leading third-party portals such as Plumtree, Microsoft SharePoint, IBM WebSphere, PeopleSoft, SAP, Epicentric, and BEA WebLogic.

The reports can be distributed:

- In a variety of output formats, including a PDF file in Adobe Acrobat format, an e-mail attachment, simple HTML text, or direct input into an Excel spreadsheet. For example, a financial analyst can receive a monthly P&L report in an Excel spreadsheet, then analyze it offline instead of re-keying data from a paper printout into the spreadsheet.
- In multiple sections, with each section sent to separate recipients. For example, a single sales report can be sent to multiple regions, with the manager of each region receiving only the information pertaining to their territory.
- To many different people in multiple versions. Unlike most other business intelligence providers, WebFOCUS needs to run a report only once to achieve multiple report delivery to multiple people.

Archiving

The WebFOCUS Report Library is a powerful archiving facility that provides organizations with the ability to store and manage financial reports for future retrieval and use. Because businesses generate large volumes of financial information on a regular basis, the ability to archive reports in a secure central storage and efficiently restore them is crucial for efficient compliance with information retention guidelines such as those of the IRS, which requires businesses to retain records for auditing purposes.

Among key features:

- Versioning—The ability to save multiple versions of the same report/document.
- Expiration—The ability to set a file's expiration date or limit versions to a specified number.
- Categorization—The ability to logically organize documents by categories and domains, so they are easy to locate and retrieve.
- Security—The ability to keep sensitive information protected from unauthorized users by defining which users can view stored content.
- E-mail Notification—The ability to send an email automatically to users to inform them when a new file is added to the archive, and provide them with a URL link to the information.

WebFOCUS Financial Report Painter

Just as financial information is a critical cornerstone of business, the WebFOCUS Financial Report Painter is a critical cornerstone of the WebFOCUS Financial Reporting Platform.

Fully integrated with the WebFOCUS Financial Reporting Platform, the robust Financial Report Painter—with its intuitive interface and matrix workpad—makes it easy for developers to create powerful, tailored financial reports that can:

- Perform calculations between row and column elements to produce reports in a row-by-row, column-by-column format.
- Post reports to an external file and pick them up at a later time for analysis—a powerful feature to develop and consolidate intermediate reports, then use the results again in a later report.
- Produce recursive reports, where the results from the end of one time period or column become the starting balance in the next—an effective way for organizations to project cash flow or other forward looking reports.
- Generate profit and loss statements, consolidations, actuals, budgets, and government-mandated financial reports easily by simple point and click action.

- Dynamically create summary financial statements from detailed financial data.
- Build financial reports from a wide range of data sources and structures, including Charts of Accounts and other hierarchies.
- Generate accurate financial reports directly from centralized, secure, and professionally maintained data sources. Organizations performing financial reporting through manual spreadsheets and other static, limited processes are at a distinct disadvantage in this highly scrutinized fast-paced business climate.
- Provide a wide range of report styling and customizing options to fit the needs of different audiences.

The Financial Report Painter is driven by WebFOCUS' powerful financial modeling language, which is capable of handling large volumes of data, volatile data relationships, and custom Charts of Accounts. These Charts of Accounts can be quickly displayed in hierarchical trees that can be compressed or expanded with a simple click of the mouse, making it easy for a wide range of users to:

- View the contents and structure of financial data.
- Drag an account from any level and drop it into a report.
- Produce rows with different levels of aggregation.
- Minimize report maintenance as a result of Charts of Accounts changes. When accounts are added, deleted or realigned, the report procedure will not need to change to reflect the restructuring.
- Dynamically consolidate data in Charts of Accounts and generate real-time financial reports.

WebFOCUS Financial Report Painter's Advantage Over SQL-based Financial Reporting Tools

Although many SQL-based reporting tools claim to be well-suited for financial reporting, evidence shows that these tools are actually limited in real-world situations. For example:

- Financial reports often require account data to be grouped and calculated by rows. SQL-based tools can only retrieve, group and calculate data by columns.
- Financial reports often require items to be listed by liquidity or some other corporate or government-defined rule. SQL-based tools can only sort data alphabetically or numerically.
- Charts of Accounts hierarchies often change. SQL-based tools refer to account data explicitly, so financial reports built with SQL-based tools require constant maintenance.
- Organizations demand that financial reporting needs get filled fast. SQL-base tools usually require involvement of IT staff, however, IT personnel typically have limited understanding of financial and accounting operations, so reports must often be modified and revised several times before they fully meet the organization's needs—slowing down valuable decision-making time and putting a burden on the organization's resources. In addition, compliance with new reporting guidelines will place additional pressure on IT staff to build reports faster and more often.

Unlike SQL-based financial reporting tools, the WebFOCUS Financial Report Painter meets and exceeds these critical business demands rapidly and efficiently:

- Retrieving, grouping, and calculating data by columns.
- Grouping and calculating account data by rows.
- Sorting data alphabetically or numerically.
- Dynamically maintaining reports as organizations' Charts of Accounts changes.

WebFOCUS Financial Report Painter's Advantage for Data Access

Another advantage of the WebFOCUS Financial Report Painter is its ability to access more than 100 data sources on some 35 platforms, including:

- Multi-dimensional cubes such as Hyperion Essbase.
- Analytical data structures, including DB2, Oracle, SQL Server, Teradata.
- Data resident on legacy systems, such as IMS, VSAM, and IDMS on the mainframe, or VAX, Unix, Tandem, As/400, and NT, among others.

This ability for organizations to leverage all their data, from legacy to data warehouse, provides a powerful edge in enterprise financial reporting that very few financial reporting solutions can provide.

Learning More About the WebFOCUS Financial Report Painter

While this chapter provides an overview of WebFOCUS enterprise financial reporting—the most complete financial reporting solution your organization needs—this is just the beginning.

The rest of this book will introduce you to Information Builders' powerful yet easy-to-use financial report building tool—the WebFOCUS Financial Report Painter.

We believe that as you read how to use the WebFOCUS Financial Painter, you will begin to see more of its values, benefits and advantages. And once you start to use it, you will learn how much more the full WebFOCUS financial reporting solution can fulfill your organization's financial reporting requirements, as well as improve the organization's financial situation.

To begin to learn how to use the WebFOCUS Financial Report Painter, and to find out more about its many features, turn to Chapter 2, *Creating Financial Reports with the Financial Report Painter*.

If you wish, you can also refer to *Appendix A, Creating Reports With Financial Reports Painter the FML Language*, to discover the powerful language that drives the Financial Report Painter.

Finally, if you would like information about reporting against hierarchical data structures, turn to *Appendix B, Describing Data for an FML Hierarchy*.

CHAPTER 2

Creating Reports With Financial Report Painter

Topics:

- Starting a Financial Report
- Designing Your Financial Report
- Running a Financial Report
- Retrieving FOR Field Values From a Data Source
- Reporting Dynamically From a Hierarchy
- Supplying Data Directly
- Performing Inter-Row Calculations
- Inserting Rows of Text
- Suppressing the Display of Rows
- Saving and Retrieving Intermediate Report Results
- Formatting Financial Reports
- Adding, Inserting, and Deleting Rows
- Editing Row Types and Properties
- Adding and Deleting Columns

The WebFOCUS Financial Report Painter, and its underlying Financial Modeling Language (FML), are designed for the special needs associated with creating, calculating, and presenting financially oriented data such as balance sheets, consolidations, profit and loss statements, budgets, and certain government mandated financial reports. These reports are distinguished from other WebFOCUS reports because calculations are inter-row as well as inter-column. And each row or line represents a unique entry or series of entries that can be aggregated directly from the input data, or calculated as some function of the data.

Using an intuitive matrix design, the Financial Report Painter enables you to model and generate real-time financial reports that handle large volumes of data, volatile data relationships, and custom charts of accounts .

The Financial Report Painter generates FML code, beginning with the FOR command that initiates the row-based orientation of financial modeling, and the OVER commands that stack the rows in the report. For details about FML syntax, see Appendix A, *Creating Financial Reports With FML Language*.

Starting a Financial Report

The Financial Report Painter is a feature of the Report Painter, from which it is accessed. You start a report in the Report Painter and complete it in the Financial Report Painter. The Report Painter retains control over report-wide and column-oriented properties of the report, while the Financial Report Painter controls the row-oriented properties. However, you can access key Report Painter dialog boxes—particularly the Report Options and Field Properties dialog boxes—directly from the Financial Report Painter, making it easy to control many report and column-based properties without switching tools.

Procedure How to Start a Financial Report

In the Report Painter:

1. Add the fields you want to include in the report, including the field that you intend to designate as your FOR field. The FOR field will allow the report to be structured on a row-by-row basis in the Financial Report Painter.

You can include vertical (BY) and/or horizontal (Across) sort fields, however, these fields are not required, and, in fact, are frequently omitted from financial reports where sorting is controlled by the placement of FOR field values in the matrix. Nevertheless, if you wish you can include both BY phrases and Across phrases in the request.
2. You may add any of the following report elements as required: headings and footings, totals, subtotals, calculated values, and images.
3. Now select the field that you want to use as the FOR field, and click the *For* button on the Report Painter Columns toolbar. A report request can only contain one FOR field.
4. Click the *Matrix* tab at the bottom of the window.

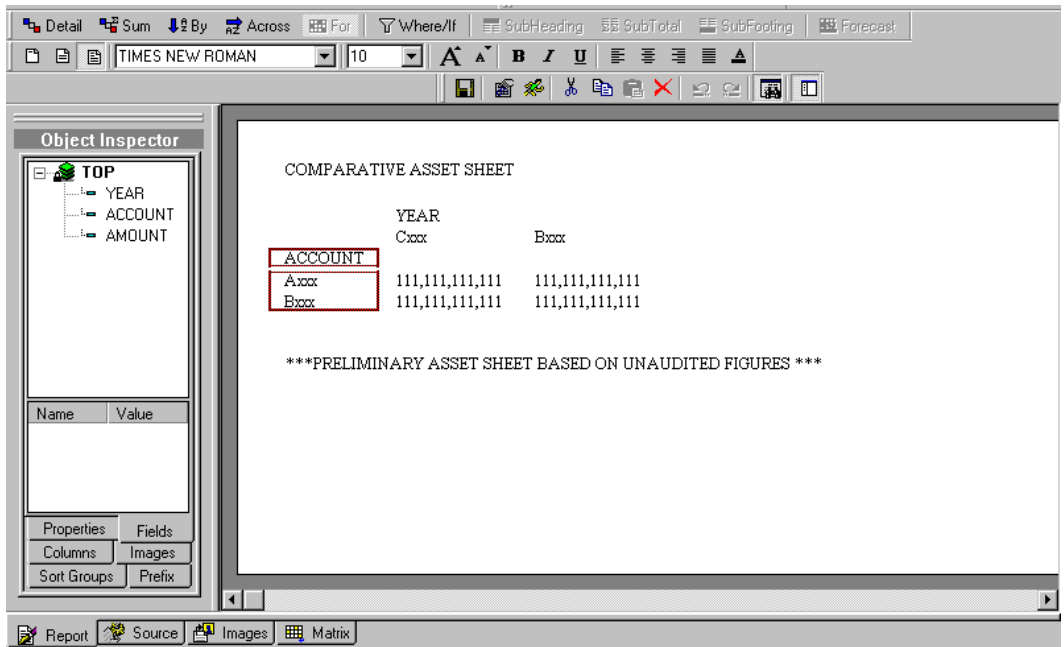
The Financial Report Painter's Design matrix opens.

Tip: If you want to return to the Report Painter, click the Report tab below the matrix or choose Show Report Painter from the View menu.

To reopen the Financial Report Painter from the Report Painter, click the Matrix tab.

Example Report Painter With FOR Field Selected

In this example, the Comparative Asset Report is open in the Report Painter. The heading and footing are defined and the report columns are specified. ACCOUNT is designated as the FOR field and the *For* button is selected on the Report Painter Columns toolbar.



The next step is to click the Matrix tab below the Report Painter window to open the report in the Financial Report Painter. For an illustration, see *Designing Your Financial Report* on page 2-4.

Designing Your Financial Report

Most of the work you will do in the Financial Report Painter occurs in the Design matrix, where you can add:

- Tags that represent values of the FOR field in the data source.
- Data that is not derived from a data source, including intermediate results from other reports that are posted to a file and picked up as data.
- Free text and blank lines.
- Underlines to set off the results of computations.
- Formulas for inter-row and intra-row calculations.

You can also format rows, columns, and cells in the Design matrix.

The following matrix contains a simple asset sheet contrasting the results of two years. (Note that, like all Financial Report Painter requests, this one originated in the Report Painter. See *Report Painter With FOR Field Selected* on page 2-3).

The screenshot shows the Financial Report Painter software interface. The main window displays the Design matrix, which is a table with the following columns: Label, Row Type, ACCOUNT, Title, and AMOUNT. The matrix contains data for various assets, including UTP, UTPAD, R3, UTPNET, R5, INV, R7, CASH, ACR, ACI, FUEL, and MAT. The AMOUNT column shows values like 11111. To the right of the matrix, a vertical list of account numbers is visible: 1000, 1010, 2000, 4000, 5000, 6000, 6500, 6600, 6900, and 7000. The interface also includes a menu bar at the top, a toolbar with various icons, and a properties panel on the left side.

Label	Row Type	ACCOUNT	Title	AMOUNT
UTP	TAG	1000	UTILITY PLANT	11111
UTPAD	TAG	1010 TO 1050	LESS ACCUMULATED DEPRECIATION	11111
R3	BAR			
UTPNET	RECAP		TOTAL PLANT-NET	UTP-UTPAD
R5	BAR			
INV	TAG	2000 TO 3999	INVESTMENTS	11111
R7	TEXT			
CASH	TAG	4000	CASH	11111
ACR	TAG	5000 TO 5999	ACCOUNTS RECEIVABLE-NET	11111
ACI	TAG	6000	INTEREST RECEIVABLE	11111
FUEL	TAG	6500	FUEL INVENTORY	11111
MAT	TAG	6600	MATERIALS AND SUPPLIES	11111

Reference **Financial Report Painter Design Matrix**

The Financial Report Painter enables you to define the elements of your report request. It consists of the matrix itself, the panel containing FOR field values from the data source, a set of toolbars, and a set of Properties check boxes.

Below the matrix are four tabs that enable you to switch quickly between the Matrix view, the Source code, the Report as it appears in the Report Painter, and the Images that may have been embedded in the report.

Matrix elements:

Row labels (R1, R2, etc.)

Are supplied by default in the first column of the matrix, where they provide unique identifiers for each report row. They become important as you place various report elements in the rows of the matrix. These labels enable you to identify rows to be used in calculations. You can also use the labels to identify rows and cells for formatting.

You can replace the default labels with more meaningful labels of your own.

Row Type

Identifies the type of information included in each row of the report. This column appears by default, but initially the values are blank. The Financial Report Painter adds the row type as you define each element in the report. Row types are: TAG, DATA, RECAP, TEXT, BAR, and PICKUP.

FOR Field

Contains the field you designated in the Report Painter. FOR is the underlying command that enables row-by-row control in a WebFOCUS financial report. A report can only contain one FOR field. The data values of the FOR field, or calculations based on those values, are stacked over each other in the matrix.

Title

Lists any titles you define for rows. If no titles are specified for TAG rows, the FOR field values are displayed as default row titles. For DATA and RECAP rows, the title is blank unless provided.

You can assign multiple data values in a single row and then assign an identifying title to the group.

If you are working with hierarchical data for which captions have been defined in the Master File, the captions are displayed in the Title column. See *Reporting Dynamically From a Hierarchy* on page 2-21.

Report fields

Each field you specify in the Report Painter (including the FOR field) appears as a column title in the matrix. If you include any sort (BY) fields, they appear before the others in the report.

Column widths are initially adjusted to fit the longest item in each column. As you type into the matrix or enter values in dialog boxes, you can adjust the column widths as required by dragging or double-clicking the boundary lines.

You can include additional columns in your report without returning to the Report Painter. For details, see *Adding and Deleting Columns* on page 2-86.

FOR field values panel:

This panel displays the values of the FOR field in a resizable, scrollable window to the right of the matrix. You can add values to the Design matrix by dragging or double-clicking. You can multi-select contiguous or non-contiguous values and drag the selected group onto the matrix, where they appear on a single line, connected by the OR command.













If you are reporting against a hierarchy, the values in the panel reflect the hierarchy tree. See *Reporting Dynamically From a Hierarchy* on page 2-21.







Design toolbars:

Financial Report Painter toolbar icons, which appear above the matrix, enable you to perform the following tasks:

- Add tag values from a data source, supplementary data values, free text, underlines, and Recap formulas to the matrix.
- Post data to a file and pick up the data that you have posted.
- Add, insert, and delete rows.
- Run financial reports.
- Change font style, size, and color, and justification for individual cells in the matrix.
- Perform standard operations such as Save, Cut, Copy, Paste, Undo/Redo.

The toolbar icons are:

Icon	Tool Tip	Description
	Report Options	Opens the Report Painter's Report Options dialog box, which control all report-wide properties.
	Run	Runs the report.
	Design mode	Toggles between the full Design matrix and a subset of the matrix columns for preview purposes.
	Show Properties of Selected Row	Shows the properties of the selected row in the dialog box in which those properties were defined.
	Tag	Opens the TAG dialog box where you can specify values and properties for the TAG row.
	Recap calculator	Opens the RECAP dialog box where you can create an expression that derives the value of the RECAP row and assign row properties.
	Text	Opens the TEXT dialog box where you can enter a row of text, or a blank row.
	Data	Opens the DATA dialog box where you can assign properties, such as title and label, to a row whose data you supply directly in the matrix (rather than retrieving it from a data source).
	Bar	Opens the BAR dialog box where you can specify an underline character.
	Pickup	Opens the PICKUP dialog box where you can assign row properties, such as title and label, and indicate the name of a file from which to pick up a row of previously posted data.
	Add a row	Adds a blank row at the bottom of the matrix.
	Insert a row	Inserts a blank row between existing rows in the matrix.

Icon	Tool Tip	Description
	Delete a row	Removes the selected row from the matrix.
	Make Recap	Creates a RECAP row based on the sum of other values in the matrix. (This option applies when no expression is required.)
	Increase/ Decrease Size	Increases or decreases font size by one measurement at a time.
	Bold, Italic, Underline	Adds emphasis using the selected style.
	Justify	Left-, Right-, Center-justifies the characters in the selected cell, or applies the default justification.
	Color	Applies the color chosen from the palette to the characters in the selected cell.

Properties Area

The Properties area above the matrix is a resizable, scrollable region in which you can assign appropriate properties to the report or to specific row types.

If an editable cell is selected in the matrix, its content is displayed in the Properties box below the check boxes. To enter values in a selected cell, type in the Properties box and press *Enter*.

Row Properties check boxes affect individual rows:

Invisible

Suppresses the display of a row.

When Exists

Suppresses the display of a row when no data exists.

Post to

Enables you to enter the ddname for a file to which you want to post a row of output for later pickup and use.

FML Properties check boxes affect the entire report:

Use Multiple Values

Enables you to add the same data value to multiple rows in the Design matrix. For example, the same value can exist as a single value in one row, as part of a range in another row, and in a calculation in a third row.

For data that has not been structured as a hierarchy, values continue to be listed in the FOR field values panel after being used in a row. Without this option, values are removed from the list after being used.

In a hierarchy, a used value remains in the FOR field values panel, where it is displayed in red to indicate that it has already been added to the matrix.

In both instances, you can reuse the value as required.

First Instance

Applies when you are reporting against a data source with shared members, such as ESSBASE, in which the same data can be defined multiple times with different hierarchy field values.

Unless this option is selected, data shared by two different parents will be counted twice in an aggregation operation. When this check box is selected, the Financial Report Painter applies the .FST prefix operator to all summed numeric columns in the report, ensuring that a shared value is only counted once.

Viewing Source Code

If you are familiar with the underlying FML syntax, as you work in the graphical tool you may wish to view the source code generated by the Financial Report Painter.

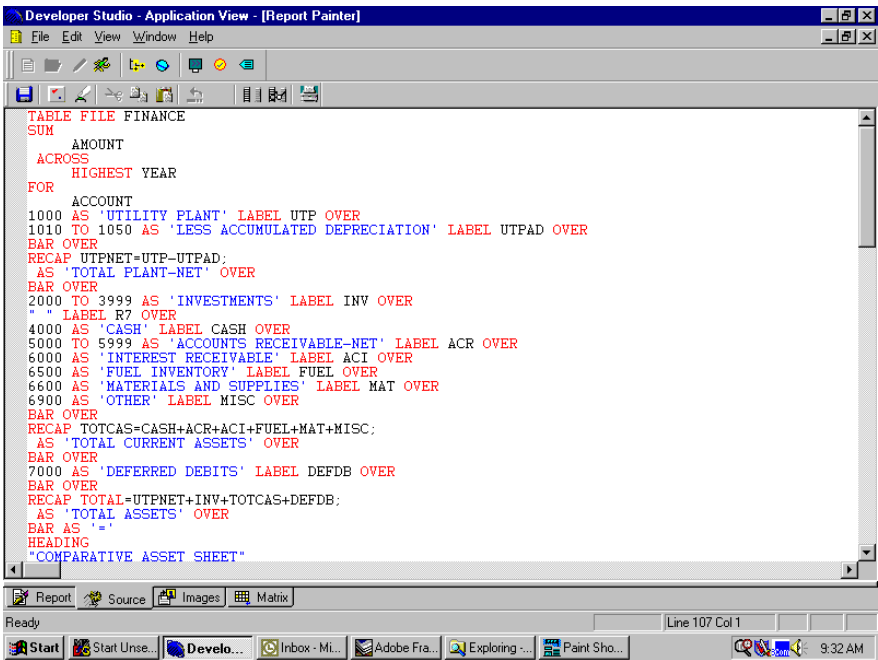
***Procedure* How to View Source Code**

To toggle from a report request in the matrix to the corresponding code, click the *Source* tab below the Financial Report Painter.

Example Viewing FML Source Code

To open the source code window, click the *Source* tab below the matrix.

The following illustration shows the source code that corresponds to the matrix for the Comparative Asset Report in *Designing Your Financial Report* on page 2-4. Notice that the code shows both the entries made in the Report Painter, as well as those made in the Financial Report Painter.



Tip: While you can edit the code that is displayed in this window, we urge you not to do so. Changes to Painter-generated code may cause problems when you try to reopen the procedure in the tool.

Previewing Your Report

You will generally work in Design mode as you build your request, but you may periodically wish to toggle to Preview mode to see the columns of the matrix that will actually be represented in the report output. For an illustration of Preview mode, see *Financial Report Painter Matrix in Preview Mode* on page 2-11.

Procedure How to Toggle Between Design and Preview Modes

Click the *Design mode* icon on the Financial Report Painter toolbar. The matrix display switches to Preview mode, which shows a subset of the information in the Design matrix.

Reference Financial Report Painter Matrix in Preview Mode

In Preview mode, the Financial Report Painter displays the following:

- In the first column, a label assigned to each row in the report—R1, R2, etc., or an explicit row label that you have supplied.
- In the second column, default titles or titles you assigned to rows in the report, representing TAG values of the FOR field, RECAP values, and other row types.
- In subsequent columns, any other fields you specified for the report.

Label	Title	1	
UTP	UTILITY PLANT	11111	
UTPAD	LESS ACCUMULATED DEPRECIATION	11111	
R3		-----	
UTPNET	TOTAL PLANT-NET	UTP-UTPAD	
R5		-----	
INV	INVESTMENTS	11111	
R7			
CASH	CASH	11111	
ACR	ACCOUNTS RECEIVABLE-NET	11111	
ACI	INTEREST RECEIVABLE	11111	
FUEL	FUEL INVENTORY	11111	
MAT	MATERIALS AND SUPPLIES	11111	
MISC	OTHER	11111	
R14		-----	
TOTCAS	TOTAL CURRENT ASSETS	CASH+ACR+ACI+FUEL+MAT+MISC	
R16		-----	
DEFDB	DEFERRED DEBITS	11111	

Report Source Images Matrix

Running a Financial Report

You can run a report directly from the Financial Report Painter at any point in the Design cycle. You can also run the report from the Report Painter or from the Developer Studio Explorer.

Procedure **How to Run a Financial Report**

Click the *Run* icon on either the Financial Report Painter toolbar or the Report Painter toolbar.

The report is displayed in the browser.

Example **Running a Comparative Asset Report**

The following sample report was begun in the Report Painter, completed in Financial Report Painter Design matrix, and run by clicking the *Run* icon on the toolbar. If you want to review the matrix from which this report was generated, see *Designing Your Financial Report* on page 2-4. The formatting of the output is accounted for by a default style template that was applied to the report.

COMPARATIVE ASSET SHEET		
	YEAR	
	1983	1982
UTILITY PLANT	1,430,903	1,294,611
LESS ACCUMULATED DEPRECIATION	249,504	213,225
TOTAL PLANT-NET	1,181,399	1,081,386
INVESTMENTS	818	5,639
CASH	4,938	4,200
ACCOUNTS RECEIVABLE-NET	28,052	23,758
INTEREST RECEIVABLE	15,945	10,206
FUEL INVENTORY	35,158	45,643
MATERIALS AND SUPPLIES	16,099	12,909
OTHER	1,264	1,743
TOTAL CURRENT ASSETS	101,456	98,459
DEFERRED DEBITS	30,294	17,459
TOTAL ASSETS	1,313,967	1,202,943

***PRELIMINARY ASSET SHEET BASED ON UNAUDITED FIGURES ***

Retrieving FOR Field Values From a Data Source

Tags identify the data values of the FOR field in your report. A report row can be associated with a tag that represents:

- A single data value of the FOR field.
- The aggregate of two or more data values of the FOR field.
- The aggregate of a range of data values of the FOR field.
- The aggregate of a related group of data values of the FOR field.

You can also identify Tag values that are organized in a parent/child hierarchy up to 99 levels deep and extending over many rows in the matrix. For details, see *Reporting Dynamically From a Hierarchy* on page 2-21.

The Tag button on the Financial Report Painter toolbar opens the TAG dialog box, in which you can define all of these variations.

Procedure How to Specify a Tag for a Single Data Value

In the Design matrix:

1. Select the row in which you want to place a tag value.

Note: If you wish to be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.

2. Click the *Tag* icon on the Financial Report Painter toolbar.

or

Right-click anywhere in the row (except on the label) and select *Change Type to >Tag* from the shortcut menu.

The TAG dialog box opens at the Options tab.

3. Enter a data value, or click the *Browse* button and select one from the list. Your entry is reflected in the Tags box.
4. Click *OK* on the Options tab or select the General tab to continue.

5. Notice that the Row label (R1, etc.) displays automatically in the Label box.
If you wish to supply an explicit row label to replace the default label (R1, etc.) on the matrix, type it into the Label input box. (The default label is retained internally.)
6. Enter a row title that you want to display on the report (optional). If you do not supply a title, the tag value is displayed.
7. Optionally, select formatting check boxes: Invisible, When Exists, Post to. For details, see *Suppressing the Display of Rows* on page 2-59 and *Saving and Retrieving Intermediate Report Results* on page 2-60.
8. Click OK.

The matrix now shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column.

Tip: You can also drag (or double-click) a tag from the FOR field values panel to a desired location on the matrix. Then right-click in the TAG row and select Row Properties from the shortcut menu to open the TAG dialog box, where you can enter a descriptive title for the tag value and/or an explicit row label.

Procedure How to Specify a Tag for Multiple Data Values

In the Design matrix:

1. Select the row in which you want to place multiple tag values.
Note: If you wish to be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.
2. Click the *Tag* icon on the Financial Report Painter toolbar.
or
Right-click anywhere in the row (except on the label) and select *Change Type to >Tag* from the shortcut menu.
The TAG dialog box opens at the Options tab.
3. Enter a data value, or click the *Browse* button and select one from the list. Your entry is reflected in the Values box.

4. Click the *Add* button to add the value to the Tags box.
5. Enter or select another data value and click the *Add* button to add the second value to the Tags box.

If you are familiar with FML syntax, note that this entry adds the OR phrase to your request.
6. Repeat steps 4- 6 for as many data values as necessary.
7. Click *OK* on the Options tab or select the *General* tab to continue.
8. Notice that the Row label (R1, etc.) displays automatically in the Label box.

If you wish to supply an explicit row label to replace the default label (R1, etc.) on the matrix, type it into the Label input box. (The default label is retained internally.)
9. Enter a row title that you want to display on the report (optional). If you do not supply a title, the tag values are displayed.
10. Optionally, select a formatting check box: Invisible, When Exists, Post to. For details, see *Suppressing the Display of Rows* on page 2-59 and *Saving and Retrieving Intermediate Report Results* on page 2-60.
11. Click *OK*.

The matrix shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column.

Tip: You can also drag (or double-click) tags from the FOR field values panel to a desired location on the matrix. (Multiple tag values may be contiguous or non-contiguous; they appear on the matrix connected by the FML keyword OR.) Then right-click in the TAG row and select Row Properties from the shortcut menu to open the TAG dialog box, where you can enter a descriptive title for the combined values and/or an explicit row label.

Procedure How to Specify a Tag for a Range of Data Values

In the Design matrix:

1. Select the row in which you want to place a range of tag values.

Note: If you wish to be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.

2. Click the *Tag* icon on the Financial Report Painter toolbar.

or

Right-click anywhere in the row (except on the label) and select *Change Type to >Tag* from the shortcut menu.

The TAG dialog box opens at the Options tab.

3. In the Value box, enter the first data value in the range, or click the *Browse* button and select one from the list.
4. In the To box, enter the last data value in the range, or click the *Browse* button and select one from the list.
5. Click the *Add* button to add the beginning and ending range values to the Tags box.
If you are familiar with FML syntax, note that this entry adds the TO phrase to your request.
6. Click *OK* or select the General tab to continue.
7. Notice that the row label (R1, etc.) displays automatically in the Label box.
If you wish to supply an explicit row label to replace the default label (R1, etc.) in the matrix, type it into the Label input box. (The default label is retained internally.)
8. Enter a row title that you want to display on the report (optional). If you do not supply a title, the tag values are displayed.
9. Optionally, select a formatting check box: *Invisible*, *When Exists*, *Post to*. For details, see *Suppressing the Display of Rows* on page 2-59 and *Saving and Retrieving Intermediate Report Results* on page 2-60.
10. Click *OK*.

The matrix now shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column.

Procedure How to Specify a Tag for a Related Group of Data Values

You can use the masking character (\$) to identify a group of data values that are identified by common characters (for example, a group of accounts beginning with the number 10). This is useful for specifying a whole group of tag values without having to name each one.

1. Select the row in which you want to place a group of tag values.

Note: If you wish to be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.

2. Click the *Tag* icon on the Financial Report Painter toolbar.

or

Right-click anywhere in the row (except on the label) and select *Change Type to >Tag* from the shortcut menu.

The TAG dialog box opens at the Options tab.

3. In the Value box, enter the common characters for the group of tags, followed by one or more masking characters (\$\$)—for example, 10\$\$ to represent 1010, 1011, 1020, 1021, 1030, 1031.
4. Click the *Add* button to add the group of tag values to the Tags box.
5. Click *OK* or select the General tab to continue.
6. Notice that the row label (R1, etc.) displays automatically in the Label box.
If you wish to supply an explicit row label to replace the default label (R1, etc.) in the matrix, type it into the Label input box. (The default label is retained internally.)
7. Enter a row title that you want to display on the report (optional). If you do not supply a title, the tag values are displayed.
8. Optionally, you can select a formatting check box: *Invisible*, *When Exists*, *Post to*. For details, see *Suppressing the Display of Rows* on page 2-59 and *Saving and Retrieving Intermediate Report Results* on page 2-60.
9. Click *OK*.

The matrix now shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column. If you do not specify a title, the tag value and masking characters are displayed.

Reference TAG Dialog Box

The screenshot shows the 'TAG' dialog box with the 'Options' tab selected. The 'General' sub-tab is active. It contains the following elements: a 'Value:' text box with a browse button (...); a 'To (optional):' text box with a browse button (...); a 'Tags:' list box containing '3100' with 'Add' and 'Delete' buttons; a 'Children' section with a dropdown menu set to 'Show only children to level...', a 'Level:' spinner box set to '1', and two checkboxes: 'Consolidate' (unchecked) and 'Display children's caption' (checked). At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Options Tab

Value

Identifies one or more data values for the TAG row.

If you are identifying a range of values, this field identifies the first data value in the range.

If you are using masking characters to specify a group of tag values with common identifying characters, enter the common and masking characters (\$\$) in the Values box.

To (optional)

Identifies the last data value in the range.

Tags

Displays the value(s) you have selected for the current TAG row.

Add button

Adds values to the Tags box.

Children

The following options appear on the TAG dialog box Options tab *only* when a parent/child relationship has been defined in the Master File being used for a report. In addition, an optional attribute in the Master File enables you to specify another field, which contains a descriptive caption for display in place of the hierarchy field values on the report. For related information, see *Reporting Dynamically From a Hierarchy* on page 2-21.

When you write a financial report against a hierarchy defined in the Master File, you can control data retrieval from the following options on a drop-down list:

Show selected item

Retrieves the value for the specified tag in the hierarchy.

Show selected item consolidated

Displays a single summary row for the numeric data values of all children.

Show only children

Displays the hierarchy starting with the first child of the specified parent, up to 99 levels deep. Each child instance is displayed over the next child instance. The parent level is *not* included in the output.

Successive levels of the hierarchy field are indented two spaces from the previous level. If you wish, you can change the spacing of these indents. See *How to Adjust Indents for Captions in a Hierarchy* on page 2-26.

Show only children to level... Level: n

Displays children up to the level you indicate. The default value is 1. Therefore, if *n* is omitted, direct children are displayed. Level 2 retrieves direct children and grandchildren. The parent level is *not* included in the output.

Show with all children

Displays the hierarchy starting with the specified parent. (Level 99 is equivalent to ALL.) The parent level is included in the output.

Show with children to level... Level: n

Displays the values for the specified parent tag and its children, up to 99 levels deep. The default value is 1. Therefore, if *n* is omitted, direct children are displayed. The parent level is included in the output.

You can refine your display further by choosing the following check box:

Consolidate

You can display a summary row for children up to the level indicated, displayed below the detail rows for the children being summed. Consolidate is supported with the Show children and Show with children options.

Display children's caption

You can display descriptive text defined in the Master File as a caption, in place of the FOR field values.

General Tab

Title

Enables you to specify a title for the TAG row; the title will appear on the report. This entry is optional.

For hierarchical data, the caption defined in the Master File is used for the title. The caption supplies descriptive text that can be used in place of the hierarchy field FOR values.

- If you add a parent to the matrix and specify *Show selected item*, the caption is applied. This value appears in the Title input line, where you can edit it.
- If you add a parent to the matrix and specify *Show selected item*, the caption is applied. This value appears in the Title input line, where you can edit it.
- If you add a parent and specify an option that also retrieves its children, captions for the children are displayed in the Title column. However, the cells are gray, indicating that you cannot change these titles. (This happens whether *Display children's caption* is on or off.)
- If no caption is specified in the Master File, the tag value is displayed as the title. However, you can provide a descriptive title for the row directly in the Title column on the matrix or in the Title input box on the General tab.

If you do not want the caption displayed, uncheck the *Display children's captions* box.

Label

Lists the default identification label (R1, R2, etc.) for the TAG row. You can replace the default row identification label with an explicit row label. (The default label is retained internally.)

In a hierarchy, each generated row is labeled with the specified label text. Children have the same label as their parents.

Formatting Options: The following options apply to individual rows in the report.

Invisible

Creates a TAG row for use in a calculation, but does not display the row on the report. For details, see *How to Suppress Rows* on page 2-59.

When Exists

Displays a row only when data exists for the tag value. By default, a TAG row is displayed even if no data is found for the tag values. The default character used to represent the missing data is a period (.). The When Exists check box enables you to override this convention. For details, see *How to Suppress Rows With No Data* on page 2-59.

Post to

Posts the output of a TAG row to a work file. This row can then be used as if were provided in a DATA row. For details, see *Saving and Retrieving Intermediate Report Results* on page 2-60.

Tip: You can also apply these formatting options from the Row Properties check boxes above the matrix

Reporting Dynamically From a Hierarchy

Hierarchical relationships between fields can be defined in a Master File and automatically displayed using the Financial Report Painter. The parent and child fields must share data values and their relationship should be hierarchical. The formats of the parent and child fields must both be numeric or both alphanumeric.

For example, suppose that:

- An employee data source contains both the employee's ID and the ID of the employee's manager.
- or
- A general ledger data source contains both an account number field and an account parent field.

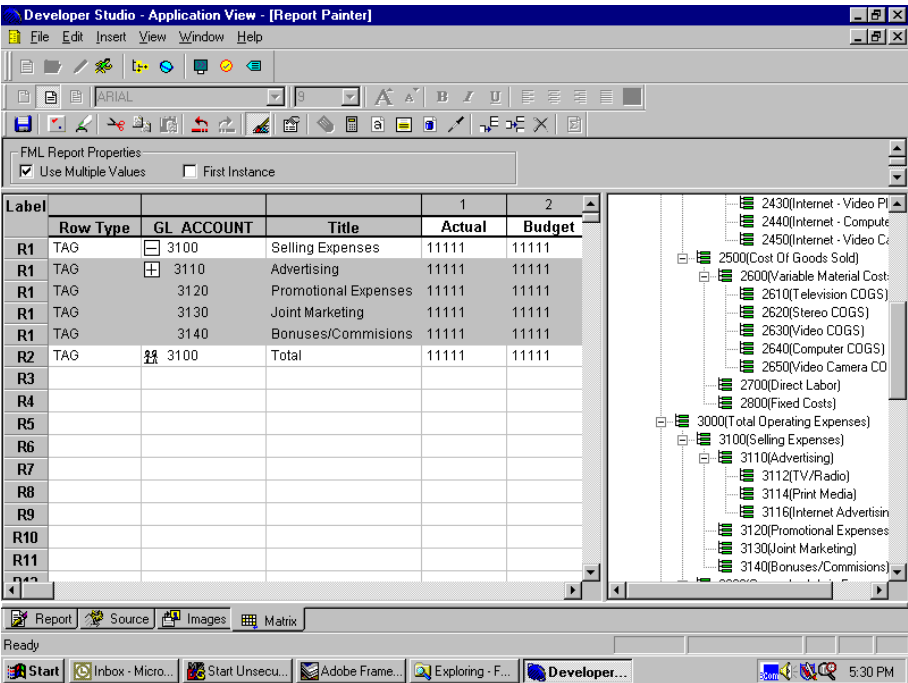
By examining these fields, it is possible to construct the entire organization chart or charts of accounts structure. However, to display the chart in a traditional row-based financial report, you would have to list the employee IDs or account numbers in the request in the order in which they should appear on the report. If an employee or account is added, removed, or transferred, you would have to change the report request to reflect this change in organizational structure.

In contrast, with FML hierarchies you can define the hierarchical relationship between two fields in the Master File and load this information into memory. The report request can then *dynamically* construct the rows that represent this relationship and display them in the report, starting at any point in the hierarchy.

For related information, see *Requirements for FML Hierarchies* in Appendix A, *Creating Financial Reports With FML Language*.

Tip: You can construct the Master File using a text editor or the graphical Master File Editor. For details, see Appendix B, *Describing Data for an FML Hierarchy*.

In the Financial Report Painter, the hierarchy defined in the Master File is reflected in the FOR field values panel to the right of the Design matrix, as shown in the following illustration. If the hierarchy fields are defined with captions in the Master File, the resulting report indents the captions proportionate to their levels in the hierarchy.



For details about how the parent/child hierarchy is represented in the Master File, see Appendix B, *Describing Data for an FML Hierarchy*.

Note: The ability to define a hierarchy in a Master File is particularly useful when you are working with a cube data structure, such as ESSBASE. However, it can be employed with other data source types.

Procedure How to Specify Tags for Data Values in a Hierarchy

In the Design matrix:

1. Select the row in which you want to place tag values that have been organized in a parent/child hierarchy. The hierarchy is reflected in the FOR field values panel to the right of the matrix.

Note: If you wish to be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.

2. Click the *Tag* icon on the Financial Report Painter toolbar.

or

Right-click anywhere in the row (except on the label) and select *Change Type to >Tag* from the shortcut menu.

The TAG dialog box opens at the Options tab. (Note that when a hierarchy has been defined in the Master File against which you are reporting, a group of specialized options, described in steps 6-8, are included in the TAG dialog box.)

3. In the Value box, enter a value from the hierarchy tree, or click the *Browse* button and select one from the list.
4. Click the *Add* button to add the selected value to the Tags box.
5. Select the value in the Tags box to activate the hierarchy fields.

6. In the Children box, indicate what level of data you want to retrieve or consolidate in the current row of the matrix:

Show selected item consolidated

Displays a single summary row for the numeric data values of all children.

Show only children

Displays the hierarchy starting with the first child of the specified parent, up to 99 levels deep. Each child instance is displayed over the next child instance. The parent level is *not* included in the output.

Successive levels of the hierarchy field are indented two spaces from the previous level. If you wish, you can change the spacing of these indents. See *How to Adjust Indents for Captions in a Hierarchy* on page 2-26.

Show only children to level... Level: n

Displays children up to the level you indicate. The default value is 1. Therefore, if *n* is omitted, direct children are displayed. Level 2 retrieves direct children and grandchildren. The parent level is *not* included in the output.

Show with all children

Displays the hierarchy starting with the specified parent. (Level 99 is equivalent to ALL.) The parent level is including in the output.

Show with children to level... Level: n

Displays the values for the specified parent tag and its children, up to 99 levels deep. The default value is 1. Therefore, if *n* is omitted, direct children are displayed. The parent level is including in the output.

7. By default, the *Display children's caption* box is checked. This enables you to display descriptive text, defined in the Master File, in place of the hierarchy field FOR values. If you do not want the caption displayed, uncheck the *Display children's captions* box.
 - If you add a parent to the matrix and specify *Show selected item*, the caption is applied. This value appears in the Title input line, where you can edit it.
 - If you add a parent and specify an option that also retrieves its children, captions for the children are displayed in the Title column. However, the cells are gray, indicating that you cannot change these titles. (This happens whether *Display children's caption* is on or off.)
 - If no caption is specified in the Master File, the tag value is displayed as the title. However, you can provide a descriptive title for the row directly in the Title column on the matrix or in the Title input box on the General tab.

8. If you chose an option that retrieves children in step 6, you can check the *Consolidate* box on the Options tabs to display a summary row for children up to the level indicated, immediately following the detail rows for the children being summed.
9. Click *OK* or select the General tab to continue.

10. Row label (R1, etc.) will automatically display in the Label box.

If you wish to supply an explicit row label to replace the default label (R1, etc.) in the matrix, type it into the Label input box. (The default label is retained internally.)

The same label applies to the parent level and all child levels.

11. Optionally, you can select a formatting check box from the General tag: The formatting check box options are:

- Invisible
- When Exists
- Post to

For details, see *Suppressing the Display of Rows* on page 2-59 and *Saving and Retrieving Intermediate Report Results* on page 2-60.

12. Click *OK*.

The matrix now shows TAG in the Row Type column for the parent values. Each parent value for children that has been retrieved is preceded by a plus sign (+) that you can click to display the children. The plus sign (+) becomes a minus sign (-) that you can click to roll up the child values.

If you entered an explicit label in the Label input box, this element will appear in the matrix.

If you selected the *Use Multiple Values* check box in the FML Report Properties area above the matrix, the used value remains available in the FOR field values panel. It is displayed in red to indicate that it has already been added to the matrix.

Tip: You can also drag (or double-click) a tag from the FOR field values panel to a desired location on the matrix. Then right-click in the TAG row and select Row Properties from the shortcut menu to open the TAG dialog box.

Procedure How to Adjust Indents for Captions in a Hierarchy

To clarify relationships within a hierarchy, the captions (titles) of values are indented at each level. If you wish to adjust the indents:

1. Choose *Indent* from the File menu. The Indent dialog box opens.
If a hierarchy has not been defined in the Master File for the data you are reporting against, the Indent option does not appear.
2. Choose one of the following radio buttons:
 - *Turn indent off*. This option left-justifies titles for values at all levels of the hierarchy.
 - *Select a value of indent*, then enter an explicit measurement to represent the number of spaces to be indented based on the number of blank spaces preceding the caption text in the Master File and the unit of measurement defined in the Page Setup dialog box (inches, centimeters, or points). For example, suppose that the caption text is preceded by two blanks and the base measurement is inches. If you enter .2, each level of values in the hierarchy will be indented .2 inches for each leading space (.2" x 2) from the previous level.

Displaying an FML Hierarchy

When reporting from an FML hierarchy, you can dynamically retrieve and display hierarchical data using two variations:

- *Show only children* displays only the children, not the parent value for those children.
- *Show with children* displays the parent and then the children.

In either case, you can show all children for the specified parent, or children to a specified level in the hierarchy, up to 99 levels deep. The default hierarchy depth is one level. To highlight these relationships visually, successive levels of the hierarchy field are indented two spaces from the previous level. (If you wish, you can change the indentations. See *How to Adjust Indents for Captions in a Hierarchy* on page 2-26.)

The hierarchy is displayed sorted by the parent field and, within parent, sorted by the hierarchy field.

When displaying a hierarchy, you can show either detail or summary data for fields other than the For field in the request. You can also replace the For field value with a descriptive caption.

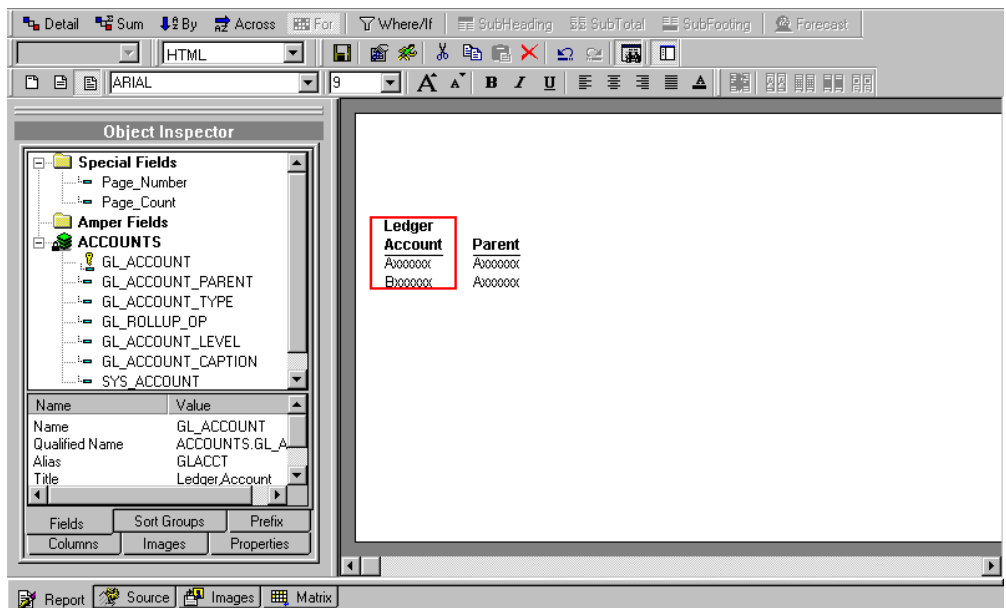
Example Displaying an FML Hierarchy With Captions

The following example displays two levels of a charts of accounts hierarchy and shows descriptive captions defined in the Master File, instead of the account numbers, in the report.

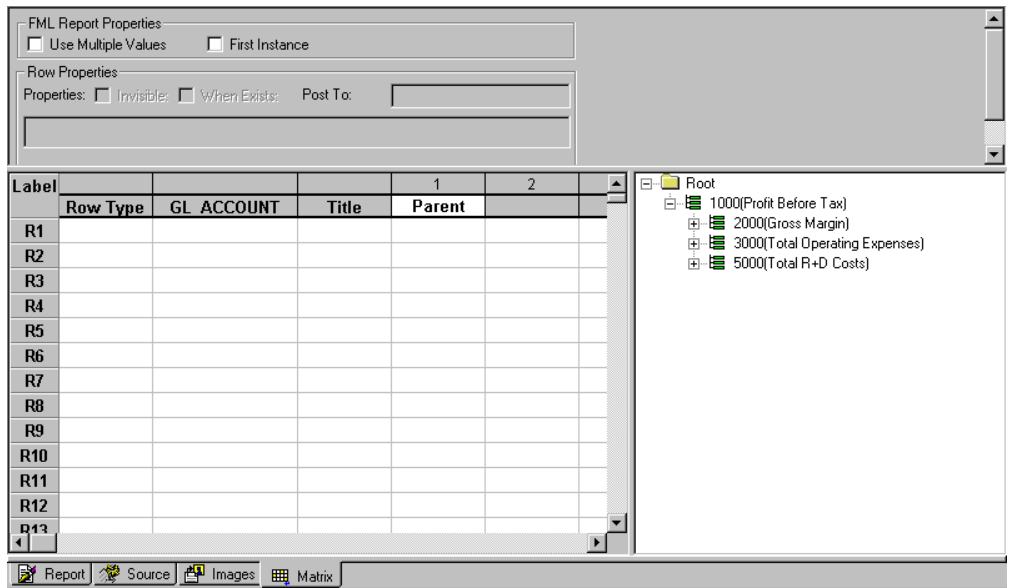
1. Create a report procedure in Developer Studio and name the file *fmlhiercaps*.
2. When prompted, choose Report Painter as your reporting tool (it is the default) and choose CENTGL as your data source. To examine this data, see *Sample Data for FML Hierarchy Examples* on page 2-44.

The Report Painter opens with CENTGL fields listed in the Object Inspector panel.

3. Drag the field GL_ACCOUNT_PARENT into the Report Painter window; the column title associated with this field is Parent. Select *Parent* and click the *Detail* button.
4. Now drag GL_ACCOUNT into the Report Painter window; the column title associated with GL_ACCOUNT is Ledger Account. Select *Ledger Account* and click the *For* button. (This will become the controlling field in your financial report.) The Report Painter looks like this:

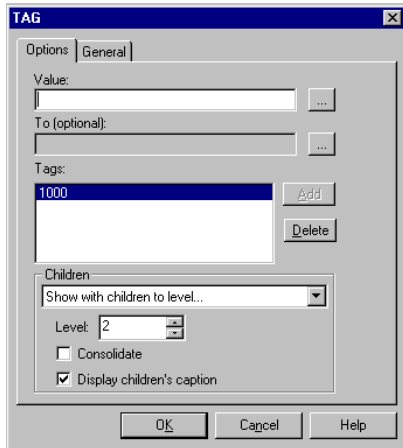


- Click the Matrix tab below the Report Painter to open the Design matrix. Note that GL_ACCOUNT (the For field) is the title of the second column and its values appear in the For field values panel at the right of the matrix. You will be populating the matrix with these values.

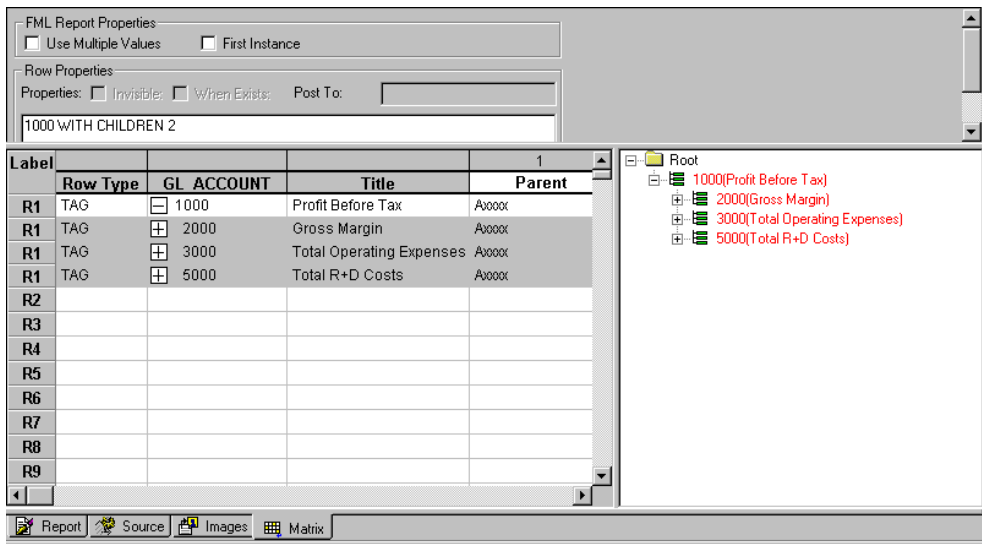


- Begin by dragging the tag 1000 from the FOR field values panel into row R1 of the matrix. The 1000 account tag appears in the GL_ACCOUNT column.
- In row R1, right-click 1000 and select Row Properties from the menu. The TAG dialog box opens, with 1000 listed in the Tags box.

8. In the Children box, choose *Show with children to level...* from the drop down list, then select 2 in the Level box to display two levels of the hierarchy, with account 1000 as the starting point (or parent level). The *Display children's captions* check box is selected by default. This will show the descriptive titles for the children, rather than their tag values (1000, 2000, etc.), on the report. The dialog box looks like this:



9. Click the plus (+) sign next to 1000 to expand the hierarchy one level. The matrix now looks like this:



If you wish, repeat the process to expand the hierarchy another level.

10. To add some quick styling, click the *Report Options* icon on the toolbar above the matrix. On the *Style* tab, click the *Style File Selection* button (bottom right of window). The *Style File Selection* dialog box opens. Select the style sheet *DefaultGrid* (*DEFFLT1.STY*) under *Style Sheet Source*. Click *OK* to confirm your choice and *OK* again to return to the matrix.
11. Click the *Run* icon on the toolbar to see the report, which lists the account numbers associated with the levels of the parent/child hierarchy. The indents for the hierarchy levels are set by default.

	<u>Parent</u>
Profit Before Tax	
Gross Margin	1000
Sales Revenue	2000
Cost Of Goods Sold	2000
Total Operating Expenses	1000
Selling Expenses	3000
General + Admin Expenses	3000
Total R+D Costs	1000
Salaries	5000
Misc. Equipment	5000

Tip: If you wanted to see the children in the hierarchy without the parent, you could choose *Show only children to level 2* in step 8, rather than *Show with children to level 2*. Without the parent line the report would look like this:

	<u>Parent</u>
Gross Margin	1000
Sales Revenue	2000
Cost Of Goods Sold	2000
Total Operating Expenses	1000
Selling Expenses	3000
General + Admin Expenses	3000
Total R+D Costs	1000
Salaries	5000
Misc. Equipment	5000

Consolidating an FML Hierarchy

The Consolidate option consolidates multiple levels of the hierarchy on one line of the report output. Consolidate can be used alone or in conjunction with the Show children or Show with children options. Consolidation is designed to work with requests that use the Sum option for fields other than the For field. It is also designed to be used with detail level data, not data that is already consolidated.

- When used *alone*, the Consolidate option aggregates the parent and children on one line of the report output, summing the numeric data values included on the line.
- When used in conjunction with *Show only children*, the Consolidate option displays one line for each child of the specified parent value. Each line is a summation of that child and all of its children. You can specify the number of levels of children to display (which determines the number of lines generated on the report output) and the depth of summation under each child. By default, only direct children will have a line in the report output and the summary for each child will include all of its children.
- When used in conjunction with *Show with children*, the Consolidate option first displays a line in the report output that consists of the summation of the parent value and all of its children. Then it displays additional lines identical to those displayed by *Show only children* plus *Consolidate*.

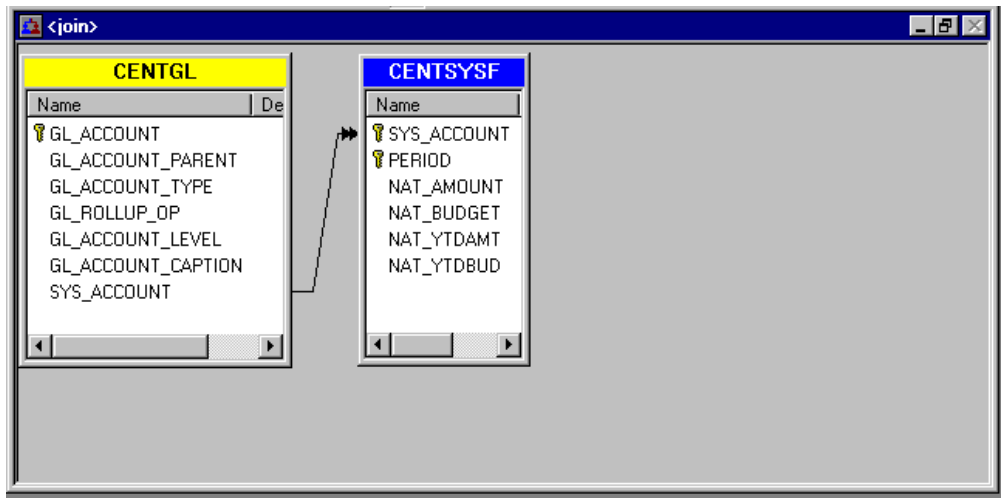
In order to use a data record in more than one line of a financial report (for example, to display both detail and summary lines or to consolidate detail data at multiple levels), select the option *Use Multiple Values* before you begin to populate the matrix.

Example **Displaying One Summary Line for an FML Hierarchy**

Data Detour: For this example, you will use two data sources: CENTGL and CENTSYSF. CENTSYSF contains detail level financial data. CENTGL defines the account hierarchy. To use the financial data with the account hierarchy, you will need to join the two data sources.

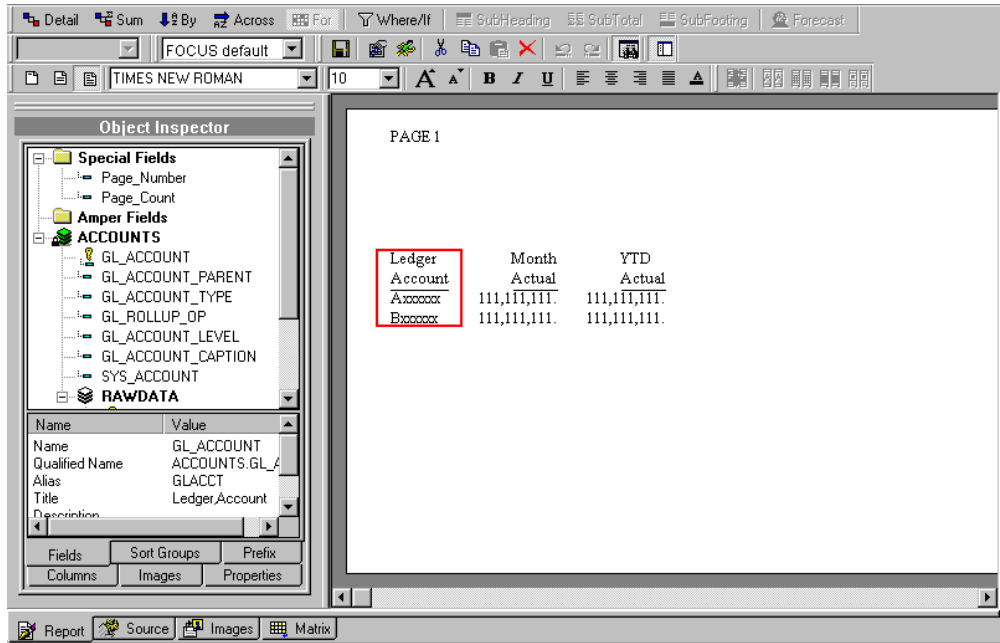
Before you complete this example, we urge you to examine the data you will be using. While you can follow the flow of this example without looking at the data, doing so will help you understand the process and the output more fully. See *Sample Data for FML Hierarchy Examples* on page 2-44. In Developer Studio, create a procedure and name the file *fmlhiersumrow*.

1. When prompted, choose Report Painter as your reporting tool (it is the default) and choose CENTGL as your host data source. The component connector window opens briefly, followed by the Report Painter, where CENTGL fields are listed in the Object Inspector panel.
2. Click the Prefix tab below the Object Inspector, right-click in the Object Inspector and choose Add New Join from the menu. When prompted, specify CENTGL as the host file. When the Join tool opens, click the *Add* icon to choose CENTSYSF as the cross-reference file. The Join is created automatically, based on the SYS_ACCOUNT field from each data source, as shown below:

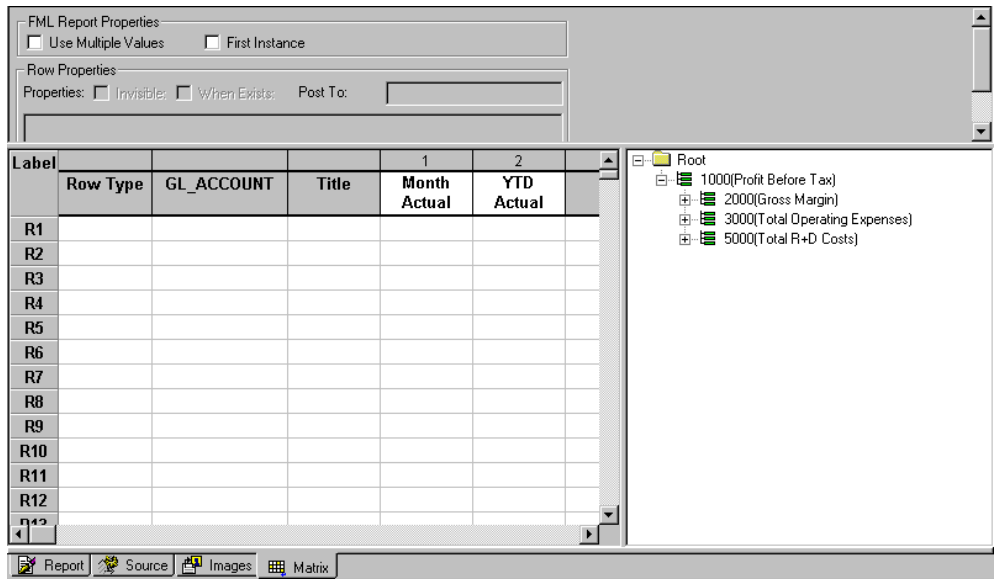


3. Run the Join and close the tool. The fields from CENTGL and CENTSYSF are now listed in the Object Inspector.
4. Drag GL_ACCOUNT (Ledger Account), NAT_AMOUNT (Month Actual), and NAT_YTDAMT (YTD Actual) into the Report Painter window.

5. Select Month Actual and click the *Sum* button; repeat for YTD Actual. Select Ledger Account and click the *For* button. (The For field will become the controlling field in your financial report.) The Report Painter now looks like this:

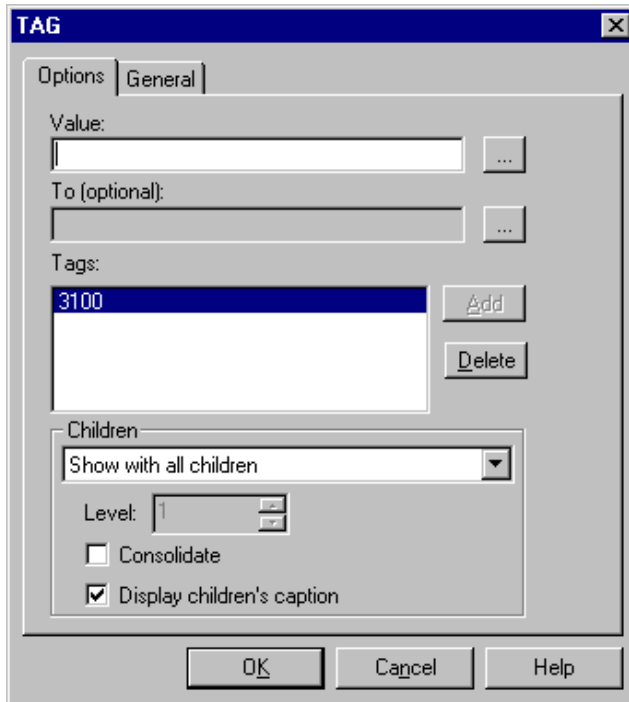


6. Click the Matrix tab below the Report Painter to open the Design matrix. Note that GL_ACCOUNT (the For field) is the title of the second column and its values appear in the For field values panel at the right of the matrix. You will be populating the matrix with these values.



7. First, click the *Use Multiple Values* check box above the matrix. This will enable you to use the same For field value more than once, to present both the detail data and the consolidation.
8. Now, drag 3100 onto the matrix in row R1. Right-click on 3100 and select Row Properties. The TAG dialog box opens, with 3100 in the Tags list.

9. In the Children box, select *Show with all children* and accept the default to *Display children's captions*. This will generate detail data for the parent value (3100) and all of its children, represented on the report by their descriptive captions rather than their account numbers.



10. To add an underline below the detailed numeric data, right-click in row R2, select *Change Row type to*, and click the radio button for *Bar*. The BAR dialog box opens. Accept the default underline character.

11. Drag 3100 into the matrix again. (Notice that 3100 and its children are now red to indicate that they have already been used in the matrix.) Right-click and choose Row Properties. The TAG dialog box opens. This time, in the Children box, select *Show selected item consolidated* to generate one summary line for Selling Expenses and all of its children.

TAG

Options | General

Value:

To (optional):

Tags:

3100

Children

Show selected item consolidated

Level:

☐ Consolidate

☒ Display children's caption

The matrix now looks like this:

FML Report Properties

☒ Use Multiple Values ☐ First Instance

Row Properties

Properties: ☐ Invisible: ☐ When Exists: Post To:

Label	Row Type	GL_ACCOUNT	Title	1 Month Actual	2 YTD Actual
R1	TAG	3100	Selling Expenses	11111	11111
R2	BAR			-----	-----
R3	TAG	3100	Selling Expenses	11111	11111
R4					
R5					
R6					
R7					
R8					
R9					
R10					
R11					

Report Source Images Matrix

Root

- 1000(Profit Before Tax)
- 2000(Gross Margin)
- 3000(Total Operating Expenses)
 - 3100(Selling Expenses)
 - 3200(General + Admin Expenses)
- 5000(Total R+D Costs)

12. Click the plus (+) sign next to 3100 in row R1 to expand the hierarchy:

Label	Row Type	GL_ACCOUNT	Title	1 Month Actual	2 YTD Actual
R1	TAG	3100	Selling Expenses	11111	11111
R1	TAG	3110	Advertising	11111	11111
R1	TAG	3120	Promotional Exper	11111	11111
R1	TAG	3130	Joint Marketing	11111	11111
R1	TAG	3140	Bonuses/Commisi	11111	11111
R2	BAR			-----	-----
R3	TAG	3100	Selling Expenses	11111	11111
R4					
R5					
R6					
R7					
R8					
R9					
R10					
R11					

Root

- 1000(Profit Before Tax)
- 2000(Gross Margin)
- 3000(Total Operating Expenses)
- 5000(Total R+D Costs)

Report Source Images Matrix

Notice that Advertising has its own children, while the other children of Selling Expenses do not. Notice also that the consolidated row, R3, has no plus sign (+) or hierarchy to expand.

13. To add some quick styling, click the *Report Options* icon on the toolbar above the matrix. On the Style tab, click the *Style File Selection* button (bottom right of window). The Style File Selection dialog box opens. Select the style sheet *DefaultGrid (DEFFLT1.STY)* under Style Sheet Source. Click *OK* to confirm your choice and *OK* again to return to the matrix.

14. Click the *Run* icon in the toolbar above the matrix to generate the following output:

	Month <u>Actual</u>	YTD <u>Actual</u>
Selling Expenses	.	.
Advertising	.	.
TV/Radio	1,049,146.	2,954,342.
Print Media	244,589.	721,448.
Internet Advertising	9,542.	29,578.
Promotional Expenses	53,719.	151,732.
Joint Marketing	97,135.	289,799.
Bonuses/Commissions	100,188.	304,199.
	<hr/>	<hr/>
Selling Expenses	1,554,319.	4,451,098.

Note that only accounts with no children are populated in the detail level data source (CENTSYF), therefore, no values are displayed for Selling Expenses and Advertising. The consolidation row accurately sums the detailed data.

Tip: To display the sum of just the children, you must display the parent row, display the summary row, and use a RECAP to subtract the parent row from the sum. For example:

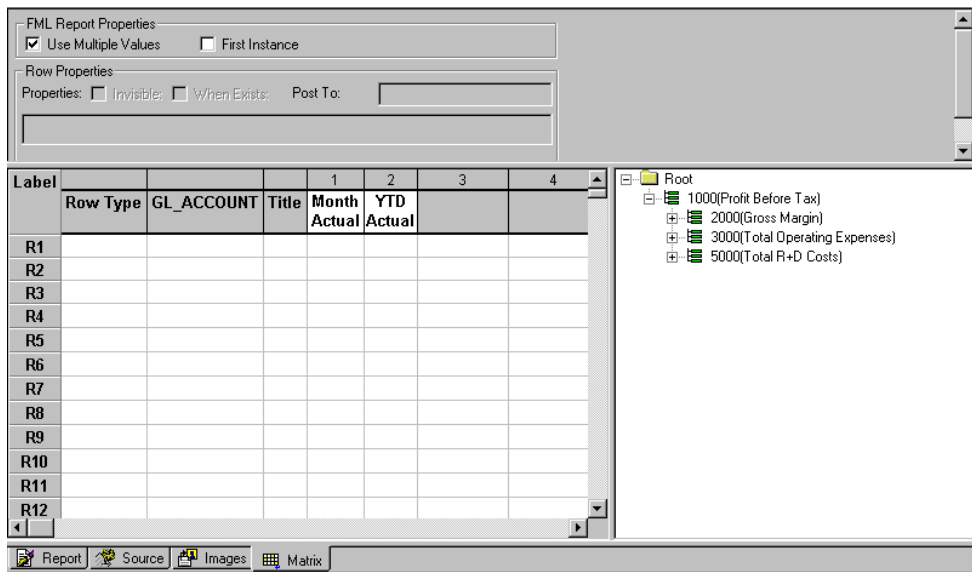
`RECAP CHILDSUM = R2-R1;`

Example Consolidating FML Hierarchy Data

1. Create a procedure called *fmlconsolidat* in Developer Studio, following steps 1-8 in the example, *Displaying One Summary Line for an FML Hierarchy* on page 2-32. Note that this example uses the data described in *Sample Data for FML Hierarchy Examples* on page 2-44.

Tip: If you wish, rather than starting from scratch, you can open the matrix created in *Displaying One Summary Line for an FML Hierarchy* on page 2-32, delete the content of the matrix, and then complete the steps that follow.

2. You should be looking at an empty matrix, with *Use Multiple Value* checked.



Drag 3100 onto the matrix in row R1. Right-click on 3100 and select Row Properties. The TAG dialog box opens, with 3100 in the Tags list.

3. In the Children box, select *Show with all children* and accept the default to *Display children's captions*. This will generate detail data for the hierarchy, starting with account 3100. Descriptive captions, rather than account numbers, will display in the output.
4. To separate the detailed output in the top section of the report from the consolidated output in the bottom section, you will add a blank row. Right-click in row R2 and select Change Row Type to *Text*. The TEXT dialog box opens. Leave the input box blank to specify a row with no content and click OK.
5. To underline the numeric columns, right-click in row R3, and select Change Row Type to *Bar*. Accept the default underline character and click OK.

6. Repeat step 4 to make row R4 a blank line as well. The matrix now looks like this:

Label	Row Type	GL_ACCOUNT	Title	1 Month Actual	2 YTD Actual	
R1	TAG	3100	Selling Expenses	11111	11111	
R2	TEXT					
R3	BAR			-----	-----	
R4	TEXT					
R5						
R6						
R7						
R8						
R9						
R10						
R11						
R12						

Root

- 1000(Profit Before Tax)
- 2000(Gross Margin)
- 3000(Total Operating Expenses)
 - 3100(Selling Expenses)
 - 3110(Advertising)
 - 3120(Promotional Expenses)
 - 3130(Joint Marketing)
 - 3140(Bonuses/Commissions)
 - 3200(General + Admin Expenses)
- 5000(Total R+D Costs)

7. To create a consolidated line for the parent account (3100) and each direct child, drag account 3100 into row R5, right-click 3100, and choose Row Properties. In the TAG dialog box, select *Show with all children* and click the Consolidate check box.

TAG

OptionsGeneral

Value:

...

To (optional):

...

Tags:

3100

Add

Delete

Children

Show with all children

Level: 1

☒ Consolidate

☒ Display children's caption

OK

Cancel

Help

The matrix now looks like this:

Label	Row Type	GL_ACCOUNT	Title	1
R1	TAG	3100	Selling Expenses	1111
R2	TEXT			
R3	BAR			
R4	TEXT			
R5	TAG	3100	Selling Expenses	1111
R6				
R7				
R8				
R9				
R10				
R11				
R12				
R13				

Root

- 1000(Profit Before Tax)
 - 2000(Gross Margin)
 - 3000(Total Operating Expenses)
 - 3100(Selling Expenses)
 - 3110(Advertising)
 - 3120(Promotional Expenses)
 - 3130(Joint Marketing)
 - 3140(Bonuses/Commissions)
 - 3200(General + Admin Expenses)
 - 5000(Total R+D Costs)

Report Source Images Matrix

- Before you run the report, limit the data to be retrieved to the period from 2002-2003. To do this, click the Report menu at the top of the screen and choose the Where/If tab. The Expression Builder opens. Using any method you choose, create the following expression:

`PERIOD EQ '2002/03'`

- Now click the *Run* icon.

In the following output, notice that:

The *top* portion shows the detail level data.

The *bottom* portion shows the consolidated data. In the consolidated portion of the report:

- There is one line for the parent that is the sum of itself plus all of its children to all levels.
- There is one line for each direct child of account 3100 (Selling Expenses): Advertising, Promotional Expenses, Joint Marketing, and Bonuses/Commissions.
- The line for Advertising is the sum of itself plus all of its children. If it had had multiple levels of children, they would all have been added into the sum. The other direct children of 3100 did not themselves have children, so the sum on each of those lines consists of only the parent value.

| | Month
<u>Actual</u> | YTD
<u>Actual</u> |
|----------------------|------------------------|----------------------|
| Selling Expenses | . | . |
| Advertising | . | . |
| TV/Radio | 1,049,146. | 2,954,342. |
| Print Media | 244,589. | 721,448. |
| Internet Advertising | 9,542. | 29,578. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |
| <hr/> | | |
| Selling Expenses | 1,554,319. | 4,451,098. |
| Advertising | 1,303,277. | 3,705,368. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |

Tip: If you wanted to see the children in the hierarchy without the parent, you could choose *Show only children*, rather than *Show with children*. Without the parent line the report would look like this:

| | Month | YTD |
|----------------------|---------------|---------------|
| | <u>Actual</u> | <u>Actual</u> |
| Advertising | . | . |
| TV/Radio | 1,049,146. | 2,954,342. |
| Print Media | 244,589. | 721,448. |
| Internet Advertising | 9,542. | 29,578. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |

| | | |
|----------------------|------------|------------|
| Advertising | 1,303,277. | 3,705,368. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |

What makes this reporting dynamic?

Suppose that you run this report weekly. If you ran the report today, and tomorrow another account were added as a child of 3100, next week's report would automatically reflect the change in the hierarchy, as well as changes to the data, based on the versatile parent/child declaration in the Master File. No adjustment would be needed to keep either the Master File or the report request up to date.

Reference Sample Data for FML Hierarchy Examples

The CENTGL Master File contains a charts of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field:

```
FILE=CENTGL          , SUFFIX=FOC
SEGNAME=ACCOUNTS, SEGTYPE=S01
FIELDNAME=GL_ACCOUNT,          ALIAS=GLACCT,  FORMAT=A7,
                                TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT,    ALIAS=GLPAR,   FORMAT=A7,
                                TITLE=Parent,
                                PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE,      ALIAS=GLTYPE,  FORMAT=A1,
                                TITLE=Type, $
FIELDNAME=GL_ROLLUP_OP,         ALIAS=GLROLL,  FORMAT=A1,
                                TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL,     ALIAS=GLLEVEL, FORMAT=I3,
                                TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION,    ALIAS=GLCAP,   FORMAT=A30,
                                TITLE=Caption,
                                PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT,          ALIAS=ALINE,   FORMAT=A6,
                                TITLE='System,Account,Line', MISSING=ON, $
```

The CENTSYSF data source contains detail-level financial data. This is unconsolidated financial data for a fictional company, CenturyCorp. It is designed to be separate from the CENTGL database as if it came from an external accounting system. It uses a different account line system (SYS_ACCOUNT) which can be joined to the SYS_ACCOUNT field in CENTGL. Data uses "natural" signs (expenses are positive, revenue negative).

```
FILE=CENTSYSF        , SUFFIX=FOC
SEGNAME=RAWDATA      , SEGTYPE=S2
FIELDNAME = SYS_ACCOUNT , , A6 , FIELDTYPE=I
                                TITLE='System,Account,Line', $
FIELDNAME = PERIOD      , , YYM , FIELDTYPE=I, $
FIELDNAME = NAT_AMOUNT  , , D10.0 , TITLE='Month,Actual', $
FIELDNAME = NAT_BUDGET  , , D10.0 , TITLE='Month,Budget', $
FIELDNAME = NAT_YTDAMT  , , D12.0 , TITLE='YTD,Actual', $
FIELDNAME= NAT_YTDBUD   , , D12.0 , TITLE='YTD,Budget', $
```

You can create an FML hierarchy in a Master File using a text editor or the graphical Master File Editor. For details, see Appendix B, *Describing Data for an FML Hierarchy*.

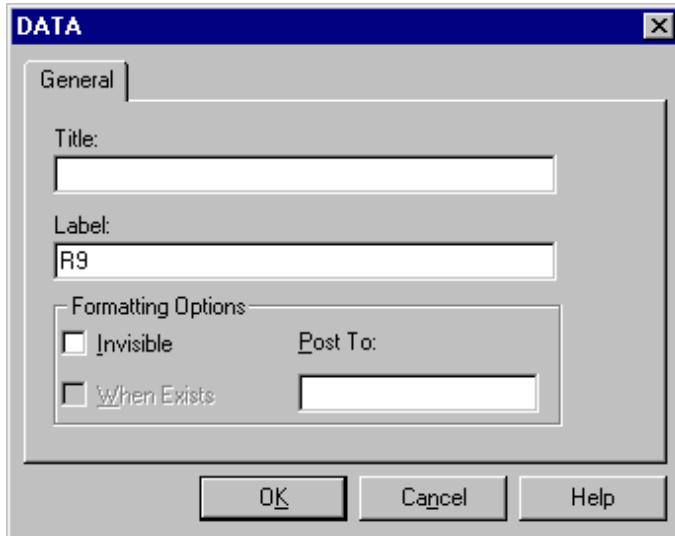
Supplying Data Directly

In certain cases, you may need to include some additional constants (for example, exchange rates, inflation rates, etc.) in your model. Not all data values for the model have to be retrieved from the data source. Using the Financial Report Painter, you can supply data directly in the request. For related information, see *Saving and Retrieving Intermediate Report Results* on page 2-60.

Procedure How to Add Data Values Directly to the Report

1. Click the *Data* icon on the Financial Report Painter toolbar.
or
Right-click anywhere in the row (except on the label) and select *Data* from the shortcut menu.
The DATA dialog box opens at the General tab.
2. In the General tab, you can:
 - Type a row title.
 - Supply an explicit row label to replace the default label on the matrix. (The default label is retained internally.)
 - Select formatting options: Invisible and/or Post to. See *DATA Dialog Box* on page 2-46.These entries are optional.
3. Click *OK* to record your entries.
4. DATA appears in the Row Type column. Type the data values (constants) in the appropriate cells in the matrix.

Reference DATA Dialog Box



General Tab

Title

Enables you to specify a title for the DATA row; the title will appear on the report. This entry is optional.

Label

Lists the default label for the DATA row. You can replace the default row identification label with an explicit row label. (The default label is retained internally.)

Formatting Options:

The following options apply to individual rows in the report.

Invisible

Creates a DATA row for use in a calculation, but does not display the row on the report. For details, see *Suppressing the Display of Rows* on page 2-59.

Post to

Posts the output of a DATA row to a work file. This data can then be used as though it were provided in a DATA row. For details, see *Saving and Retrieving Intermediate Report Results* on page 2-60.

Tip: You can also apply these formatting options from the Row Properties area above the matrix.

Performing Inter-Row Calculations

The Recap operation in the Financial Report Painter enables you to perform calculations on data in the rows of a report to produce new rows. Since these calculations are performed on rows, each row referenced in the calculation must be uniquely identified either by the default label assigned by Financial Report Painter or a label you assign to the row.

To initiate a Recap operation, you must supply the format of the value that will receive the result of the calculation, and an expression that defines the calculation you wish to perform. Initially, the default row label (R1, etc.) serves as the identifying label for the calculated value. However, it is good practice to provide a more descriptive label for reference in other calculations.

In the Financial Report Painter, you can type an expression directly into the RECAP dialog box or use a tabbed dialog box to formulate your expression.

Note: If you want to create a Recap row that summarizes the data in existing rows and does not require an expression, you can use a simple alternative procedure. For details, see *How to Create a Recap Summary Row* on page 2-56.

Procedure How to Create a Recap Expression in Financial Report Painter

In the Design matrix:

1. Select the row in which you want to perform the calculation.
2. Click the *Recap icon* on the Financial Report Painter toolbar.

or

Right-click anywhere in a blank row (except on the label) and select *Change Type to > Recap* from the shortcut menu.

The RECAP dialog box opens at the Options tab.

3. You can perform a Recap for a single column, a range of columns, columns at specified intervals, or for all columns in a row. To perform a Recap calculation and display output for:
 - **A single column.** Select a column from the Columns drop-down list.
 - **A range of columns.** Enter the number of the first column in the *From Column* box and the number of the last column in the *To Column* box.
 - **Columns with an incremental interval.** Enter values in the *From Column* and *To Column* boxes, then enter an increment number. For example, type 2 for every other column, 3 for every third column, and so on.
 - **All columns in a row.** Click *Apply to row*. (The Columns box is disabled.)
4. Assign a format to the Recap output field. You can either enter the format in the Format box or click the *Format* button to open the Format dialog box. For details, see *Format Dialog Box on page 2-53*.
5. Type a Recap expression in the input box, and go to step 10.

or

Click the *Advanced* button to expand the Recap Options tab, exposing features that will assist you in building an expression that may be composed of numbers and operators (numeric, alphanumeric, Boolean, and conditional), built-in functions, sort fields, and labels. The expression you create is displayed on the Options tab box as you make your selections. This process is described in steps 6 through 9.

6. Use the Recap calculator to compose the expression that WebFOCUS will evaluate. Click numbers plus the operators you need to create numeric, alphanumeric, date, logical, and conditional expressions. For details, see Appendix B, *Writing Expressions*, in the *Creating Reports With Graphical Tools* manual.

You can also add parenthesis and single quotation marks and designate uppercase and lowercase as required by clicking the buttons below the Expression window.

7. If you wish to include a built-in function in the expression, select the function from the drop-down list. The function is added to the Expression window, along with placeholders for the required arguments. Type appropriate values over the argument placeholders, or use the tools provided to select the values you need. For details, see the *Using Functions* manual.
8. If you wish to include a row label in your expression, click the label in the Labels drop-down list. (Note that the list shows default row labels unless you have defined explicit labels.)
9. If you wish to include a vertical sort (BY) field in your expression, click the field in the By Fields drop-down list.

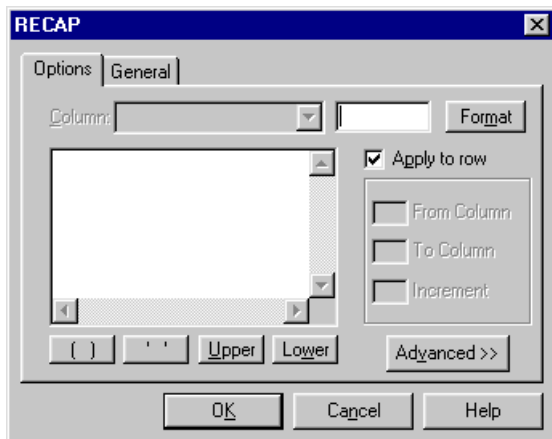
Tip: You can use a sort field value in a RECAP command to allow the model to take different actions within each major sort break. For example, in a request sorted by REGION, the following calculation would compute a non-zero value only for the EAST region:

```
IF REGION EQ 'EAST' THEN .25*CASH ELSE 0;
```

10. Click the General tab and type the title for the Recap calculation that you want to display on the report. Though optional, this entry is generally included in financial reports.
11. Notice that the row label (R1, etc.) displays automatically in the General tab's Label box. This label is used to reference the result of the Recap calculation. If you wish to supply an explicit row label to replace the default label on the matrix, type it into the Label box. (The default label is retained internally.) Though optional, this entry is generally included in financial reports.
12. Click *OK* to complete the expression.

RECAP appears in the Row Type column of the matrix, along with the expression, label, and title you define.

Reference **RECAP Dialog Box**



Options Tab

Several options in this dialog box enable you to specify the column or columns to which you want to apply a Recap formula:

Column

Identifies a single column that will contain the Recap formula; this is also the column in which the Recap output will be displayed on the report. You can choose the column from the drop-down list.

Apply to row

Indicates that the same Recap formula is to be used for all columns in the row. (The Columns box is disabled when you select Apply to row.)

From Column/To Column

Defines a range of columns to which the Recap formula is to be applied.

Increment to

Indicates an interval of columns in the range that should have the Recap formula applied. For example, a value of 2 will apply the formula to every other column.

Note: Although not directly supported by options in the Recap dialog box, there are a number of other ways in which you can identify the columns to which you want to apply the Recap calculation: column address, column value, and cell notation. To use these features, you must type the required syntax directly into the expression box in the Recap dialog box. For details, see Appendix A, *Creating Financial Reports With FML Language*.

Format box

Identifies the field type, field length, and display options that will be used for the Recap output. The field type can be alphanumeric, numeric, or date.

Format button

Opens the Format dialog box, which is designed to help you choose the format for your Recap output. See *Format Dialog Box on page 2-53*.

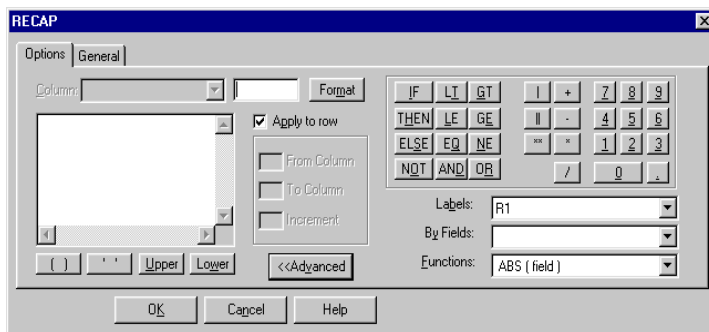
Expression box

Displays the expression used to generate the Recap output.

Type the expression, or click the *Advanced* button to expand the Options tab, exposing features that will help you create the expression.

Advanced>> button

Expands the Options tab to make it easy to add functions, numbers, operators, labels, and sort fields to your expression. Your selections are reflected in the Expression box.



Calculator

Provides numbers and operators that you can use to create numeric, alphanumeric, Boolean, and conditional expressions. Click the desired number or operator to add it to the formula in the Expression box.

Four additional buttons below the Expression box enable you to:

- Enclose a value in parentheses. Click the () button to add parenthesis.
Parentheses affect the order in which WebFOCUS performs the specified operations. For information on when to use parentheses, see Appendix B, *Writing Expressions*, in the *Creating Reports With Graphical Tools* manual.
- Enclose alphanumeric or date literals in single quotation marks. Click the ' ' button to add quotation marks.
- Convert entries in the expression box to upper or lowercase. Click the *Upper* or *Lower* button to specify case. Note that field names are case sensitive.

Labels

Lists the labels (default or explicitly defined) that you can use in your expression. Select the label you want to add from the drop-down list. Your selection is added to the formula.

By Fields

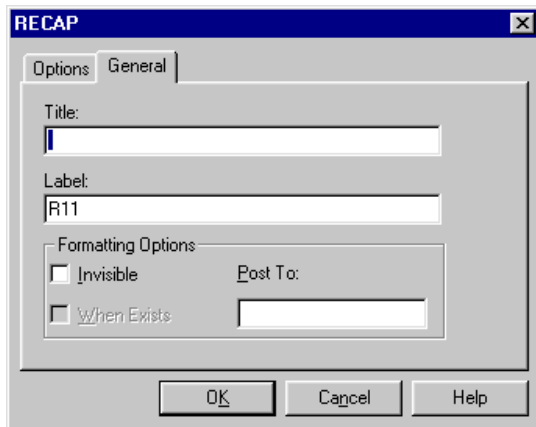
Lists the vertical sort (BY) fields that you can use in your expression. Select the fields you want to add from the drop-down list. Your selection is added to the formula.

Functions

Lists all available built-in functions that you can add to your expression. (A function is a program that returns a value.)

Select a function from the drop-down list. Then, in the Expression box, highlight each argument and substitute the value or field name you wish to use by typing or by selecting items from other drop-down lists. For details, see the *Using Functions* manual.

General Tab

The image shows a screenshot of a software dialog box titled "RECAP". It has two tabs: "Options" and "General", with "General" currently selected. Inside the "General" tab, there are three main input areas. The first is labeled "Title:" and has an empty text box. The second is labeled "Label:" and has a text box containing "R11". The third is a "Formatting Options" section containing two checkboxes: "Invisible" and "When Exists", both of which are unchecked. To the right of these checkboxes is a "Post To:" label followed by an empty text box. At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

Title

Enables you to specify a title for the Recap calculation; this title will appear on the report.

Label

Lists the default label for the Recap row. You can replace the default row identification label with an explicit row label that you want to reference on the left hand side of an expression. You can think of this label as a calculated value or as a variable that holds the result of the Recap calculation. It can also be used as part of the expression in subsequent Recap formulas.

Formatting Options

The following options apply to individual rows in the report.

Invisible

Creates a Recap expression for use in a calculation, but does not display its results on the report.

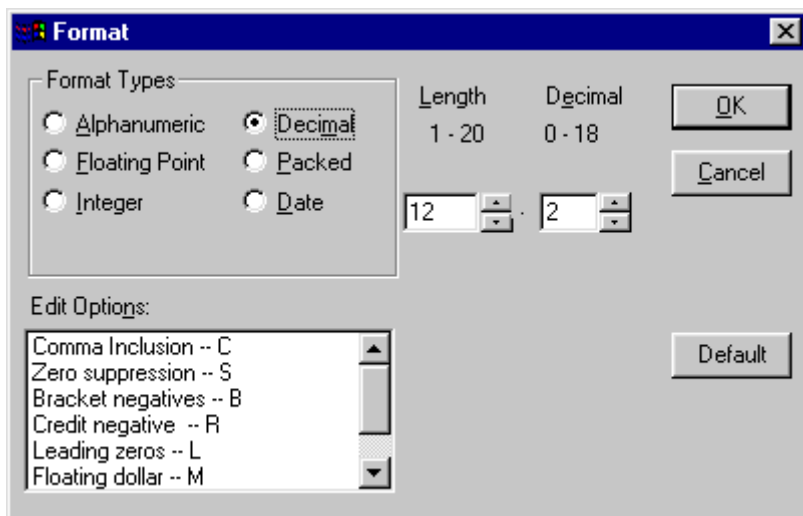
Post to

Posts the output of a Recap line to a work file. This line can then be used as though it were provided in a DATA row. For details, see *Saving and Retrieving Intermediate Report Results* on page 2-60.

Tip: You can also apply these formatting options from the Row Properties check boxes above the matrix.

Reference **Format Dialog Box**

You can use the Format dialog box to define the format of the output to be generated by a Recap calculation.



Click the *Format* button to access the Format dialog box. The Format dialog box has the following fields/options:

Format Types

Specifies the format of the current field: alphanumeric, numeric, or date.

Length

Specifies the length, in characters, of a field. Enter a number in the Length box, or click the arrow buttons to specify a number.

| | |
|--|-----------------------|
| Alphanumeric | 1-256 (default, 20) |
| Floating Point | 1-9 (default, 7.2) |
| Integer | 11 (default, 5) |
| Decimal | 20.18 (default, 12.2) |
| Packed | 33.31 (default, 12.2) |
| Note: For numeric fields, include the decimal place in the length. | |
| Date | default, MDYY |

Decimal

Specifies the number of decimal places to the right of the decimal point in a Decimal, Packed Decimal, or Floating Point field.

Edit Options

Add display options to a numeric or date field’s format to control how the field will be shown in reports.

With Field

When checked, displays the real fields from the specified Master File. You can then select the real field that will be associated with the temporary field.

Date

Assign a date format to values that represent a date or date component.
Assign century options, the two-digit century value and the two-digit base year, to control the century values of a field.

Example Calculating Recap Expressions in the Financial Report Painter

This example shows two Recap expressions, with the second using values derived in the first.

The screenshot shows the Financial Report Painter interface. At the top, there are tabs for 'Report', 'Source', 'Images', and 'Matrix'. Below the tabs, there are two main sections: 'FML Report Properties' and 'Row Properties'. The 'FML Report Properties' section has two checkboxes: 'Use Multiple Values' and 'First Instance'. The 'Row Properties' section has three checkboxes: 'Invisible', 'When Exists', and 'Post To:'. Below these sections, there is a text box containing the expression '100*TOTCASH(1)/TOTCASH(2) - 100'. The main area of the interface is a table with the following columns: 'Label', 'Row Type', 'ACCOUNT', 'Title', '1', and '2'. The table contains the following rows:

| Label | Row Type | ACCOUNT | Title | 1 | 2 |
|---------|----------|---------|-----------------|--------------|---------------------------------|
| R1 | TAG | 1010 | CASH ON HAND | 11111 | 11111 |
| R2 | TAG | 1020 | DEMAND DEPOSITS | 11111 | 11111 |
| R3 | TAG | 1030 | TIME DEPOSITS | 11111 | 11111 |
| R4 | BAR | | ----- | ----- | ----- |
| TOTCASH | RECAP | | TOTAL CASH | R1 + R2 + R3 | R1 + R2 + R3 |
| R6 | TEXT | | | | |
| GROCASH | RECAP | | CASH GROWTH(%) | | 100*TOTCASH(1)/TOTCASH(2) - 100 |
| R8 | | | | | |
| R9 | | | | | |
| R10 | | | | | |
| R11 | | | | | |
| R12 | | | | | |

On the right side of the table, there are two values: 1100 and 1200. At the bottom of the interface, there are four tabs: 'Report', 'Source', 'Images', and 'Matrix'.

The first RECAP expression calculates TOTCASH (as TOTAL CASH) by adding the values in rows 1 to 3. The results of the calculation will appear in all columns; this is designated using the *Apply to Row* check box on the RECAP Options tab. See *RECAP Dialog Box* on page 2-50.

The second RECAP expression calculates GROCASH (as CASH GROWTH (%)):


- TOTCASH(1) refers to total cash in column 1.
- TOTCASH(2) refers to total cash in column 2.
- The resulting calculation will appear in column 2 (LAST_YR) of the row labeled CASH GROWTH(%); this is designated in the Columns box on the RECAP Options tab.

The output is:

| | CURRENT | LAST |
|-----------------|-------------|-------------|
| | <u>YEAR</u> | <u>YEAR</u> |
| CASH ON HAND | 8,784 | 7,216 |
| DEMAND DEPOSITS | 4,494 | 3,483 |
| TIME DEPOSITS | 7,961 | 6,499 |
| | | |
| TOTAL CASH | 21,239 | 17,198 |
| | | |
| CASH GROWTH(%) | | 23.50 |

Procedure **How to Create a Recap Summary Row**

To create a row in which existing data in other rows is summarized:

- 1. Place your cursor in a column that contains numeric data, then highlight the values you want to sum. The values to be added may be contiguous or non-contiguous.
- 2. Click the *Make Recap* icon  on the Design toolbar.
- 3. A new RECAP row appears in the Row Type column, below the last row in the matrix. The formula representing the sum of highlighted values (for example, R1 + R2 +R3) is displayed in the new row, in the column that contained the values.

Tip: Use this technique when the calculation you wish to perform is a simple addition of values that does not require an expression. For an illustration of this technique, see *Emphasizing a Row Using Border Lines* on page 2-72.

Inserting Rows of Text

You can add ad hoc text in a row of a financial report. This includes the ability to add blank rows (designated as text) to improve the appearance of the report. Your entry is identified as TEXT in the Row Type column.

Procedure How to Add Text to a Financial Report

In the Design matrix:

1. Select a row in which you want to add text or create a blank row in the report.
2. Click the *Text* icon on the Financial Report Painter toolbar.

or

Right-click anywhere in the row (except on the label) and select *Change Type to > Text* from the shortcut menu.

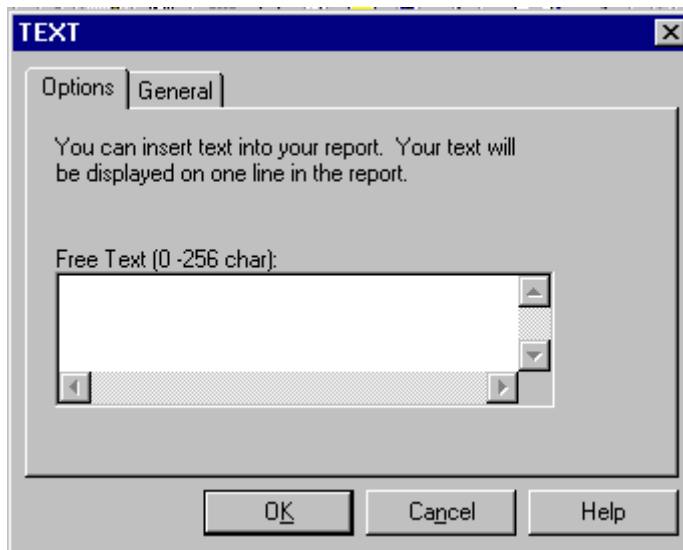
The Text dialog box opens.

3. Type up to 256 characters of text in the text box, or leave the box blank.
4. Click *OK*.

TEXT appears in the Row Type column. If you entered text, its display starts in the Title column. If you left the entry box blank, an empty TEXT row is displayed.

Tip: Once you have identified a TEXT row, you can type its content directly in the Title column on the matrix.

Reference **TEXT** Dialog Box



Options Tab

Text box

Enter up to 256 characters of text, which will appear on the matrix starting in the Title column.

Suppressing the Display of Rows

Sometimes you may want to retrieve TAG, RECAP, or PICKUP rows or add values in DATA rows solely for use in a calculation, without wishing to display those rows on the report. The Financial Report Painter enables you to mark rows as invisible for this purpose.

You may also wish to suppress the display of a TAG row if no data is found for the values. The default character used to represent the missing data is a period (.). You can override this convention, making the display of a row dependent upon the existence of data.

Procedure How to Suppress Rows

1. Select the TAG, RECAP, PICKUP, or DATA row you want to suppress.
2. Click the *Invisible* check box in under Row Properties, above the matrix.

OR

1. Right-click anywhere in the row (except on the label) and select *Row Properties* from the shortcut menu. The appropriate dialog box for the row type opens.
2. Click the General tab and choose *Invisible*.

Procedure How to Suppress Rows With No Data

To suppress the display of a TAG row with missing data:

1. Select the row you want to suppress.
2. Click the *When Exists* check box under Row Properties, above the matrix.

OR

1. Right-click anywhere in the row (except on the label) and select *Row Properties* from the shortcut menu. The appropriate dialog box for the row type opens.
2. Select the General tab and choose *When Exists*.

Saving and Retrieving Intermediate Report Results

Many reports require results developed in prior reports. This technique requires that intermediate results be stored for reuse. An example is the need to compute net profit in an Income Statement prior to calculating equity in a Balance Sheet. In the Financial Report Painter, you can save selected rows from one or more models by posting them to a work file. These rows can then be recaptured by picking them up from the work file.

The default work file is FOCPOST. Note that this is the common WebFOCUS comma-delimited file format, so you can report from it directly if a FOCPOST Master File is available. Note also that you must perform a FILEDEF on the file to assign it a logical name (ddname) prior to executing the report. You must perform a FILEDEF before running the report that posts the data, and again before running the report that picks up the data. For details on FILEDEF operations, see the *Developing Reporting Applications* manual.

Note that you cannot prepare a financial report entirely from data that you enter directly in the matrix (for example, on a DATA row); however, if you wish, you can prepare a report entirely from data that is stored in a comma-delimited work file.

Posting Data

You can save any TAG, RECAP, or DATA row by posting the output to a file. These rows can then be used as though they were provided in a DATA row.

The row will be processed in the usual manner in the report, depending on its other options, and then posted. The label of the row is written first, followed by the numeric values of the columns, each comma-separated, and ending with the terminator character of a dollar sign (\$).

For example:

| | | | |
|-----|---|--------|----------|
| AR | , | 18829, | 15954,\$ |
| INV | , | 27307, | 23329,\$ |

Procedure How to Post Data to a Work File

1. Select a TAG, RECAP, or DATA row in the Design matrix.
2. Enter the ddname of the work file in the Post To input box for Row Properties, above the matrix.

OR

1. Right-click anywhere in the row (except on the label) and select *Row Properties* from the shortcut menu. The selected Row Type dialog box opens.
2. Click the General tab and enter a ddname in the *Post To* input box under Formatting Options.

The selected row is saved in the designated work file (in comma-delimited format) from which you can then pick it up for use in another financial report.

Repeat the process for each row you wish to save to a work file (either the same file or a different one).

Note: Remember that you must perform a Filedef to assign a logical (ddname) to the work file before you execute the request.

Procedure How to Retrieve Posted Data

You can retrieve posted rows from a work file and use those rows as if they were provided in a TAG row.

Remember that you must perform a Filedef to assign a logical (ddname) to the work file before you execute the request.

From the matrix in which you want to pick up data:

1. Select the row that is to contain the picked up data. (If necessary, add or insert a blank row (see *Formatting Financial Reports* on page 2-64).
2. Click the *Pickup* icon on the toolbar.

or

Right-click anywhere in the row (except on the label) and select *Pickup* from the shortcut menu. The PICKUP dialog box opens at the General tab.

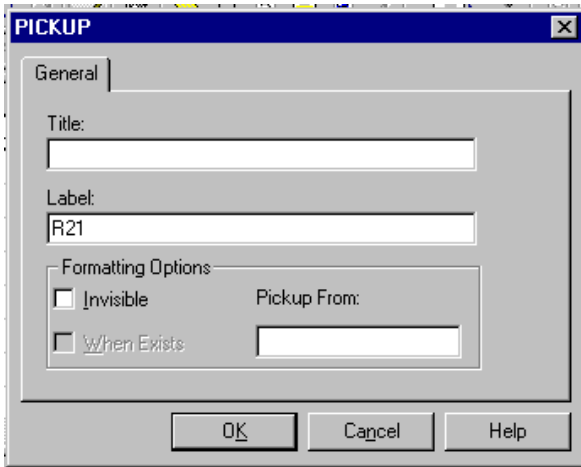
3. Enter a title for the PICKUP row in the Title box. This title will appear in the report.
4. If you wish to supply an explicit row label to replace the default label on the matrix, type it into the Label box. (The default label is retained internally.)
5. In the Pickup From box, enter the ddname for the work file that contains the posted data, followed by a space and the label of the data row you want to pick up.

6. If you wish to suppress the display of the PICKUP row, select the *Invisible* check box.
7. Click *OK*.

PICKUP appears in the Row Type column on the matrix, along with the label and title you have defined and the data values retrieved from the designated work file.

Repeat this process to include additional PICKUP rows in the report.

Reference **PICKUP Dialog Box**



General Tab

Title

Specifies a title for the picked up data; this title will appear on the report.

Label

Lists the default label for the PICKUP row. You can replace the default row identification label with an explicit row label that you want to reference in an expression.

Formatting Options

The following options apply to individual rows in the report.

Invisible

Picks up data for use in a Recap calculation, but does not display the retrieved data.

Pickup From

Is the ddname of the work file that contains the data you wish to retrieve, followed by a space and the label of the PICKUP row.

Reference **Design Matrix With Pickup Rows**

In the following matrix, two rows are designated as PICKUP:

| | Row Type | ACCOUNT | Title | CUR_YR | LAST_YR | |
|-----------|----------|--------------|---------------------|-----------------|-----------------|--|
| | | | | 1 | 2 | |
| CASH | TAG | 1010 TO 1030 | CASH | 11111 | 11111 | |
| AR | PICKUP | | ACCOUNTS RECEIVABLE | 11111 | 11111 | |
| INV | PICKUP | | INVENTORY | 11111 | 11111 | |
| R4 | BAR | | ----- | ----- | ----- | |
| CUR_ASSET | RECAP | | | CASH + AR + INV | CASH + AR + INV | |
| R6 | | | | | | |
| R7 | | | | | | |
| R8 | | | | | | |

Notice that the labels for these rows are referenced in the RECAP formula.

You can see the underlying source code, including the name of the work file to which the rows of data were posted, in the following screen.

```
TABLE FILE LEDGER
SUM CUR_YR AND LAST_YR
SUM CUR_YR NOPRINT FOR ACCOUNT
1010 TO 1030 AS 'CASH' LABEL CASH OVER
DATA PICKUP FROM BACKLG AR AS 'ACCOUNTS RECEIVABLE' LABEL AR OVER
DATA PICKUP FROM BACKLG INV AS 'INVENTORY' LABEL INV OVER
BAR AS '-' OVER
RECAP CUR_ASSET/I5C=CASH + AR + INV;

ON TABLE NOTOTAL
ON TABLE SET STYLE *
END
```

Formatting Financial Reports

You can add underlines to a financial report to set off and clarify calculations and use a variety of formatting techniques to draw attention to individual columns, rows, and cells in a financial report. As in other WebFOCUS reports, certain formatting features can be triggered by conditions that you define.

You can also drill down to another procedure or a URL from a cell or column in a financial report.

For details about conditional formatting and drill down procedures, see Chapter 3, *Creating Reports With Report Painter*, in the *Creating Reports With Graphical Tools* manual.

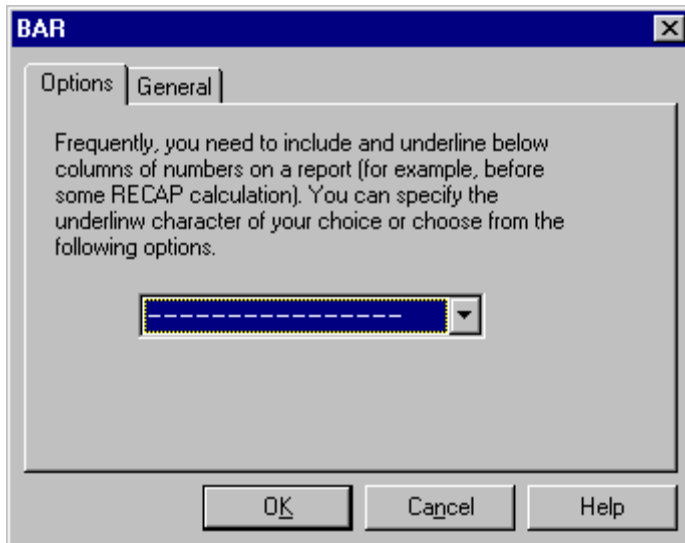
Adding Underlines

Reports that display columns of numbers frequently need to use underlines before calculations. You can specify a single or double underscore character in a BAR row.

Procedure How to Add Underlines

In the Design matrix:

1. Select the row in which you want to add an underline.
2. Click the *Bar* icon on the toolbar.
or
Right-click anywhere in the row (except on the label) and select *Change Type to...> Bar*.
The BAR dialog box opens.
3. The default BAR character is a single underscore (-). If you wish to change the underscore character, choose the double line (=) from the drop-down list on the Properties bar. (Note that in HTML reports the double line appears as a single thick line.)
4. Click OK.
5. BAR appears in the Row Type column. The selected symbol is displayed under columns that contain numeric data.

Reference BAR Dialog Box**Options tab****Underscore characters**

You can choose a single or double line. Single is the default.

Note that the double line is not clearly distinguishable in an HTML report; it appears as a thicker single line.

Formatting Columns, Rows, and Cells

You can apply a wide range of formatting options to individual columns, rows, and cells in a financial report using options on the Field Properties Style tab. You can further refine formatting:

- For columns, by identifying the column title and column data as separate objects for styling.
- For rows, by identifying the row and title as separate objects for styling.

If you style a row and then a cell within that row, the cell styling takes precedence for the cell.

If you style a column and then a cell with that column, the cell styling takes precedence for the cell.

If you style a cell, then style a row or column that the cell is in, the cell style will remain the same.

Row and cell styling options are specific to financial reports created in the Financial Report Painter. Column styling options are identical, whether applied from the Financial Report Painter or the Report Painter.

Procedure **How to Format a Column in the Design Matrix**

1. Right-click a column and choose *Options* from the shortcut menu. The Field Properties dialog box opens at the Style tab.
2. Select *Column Title and Data* as the active object, or select *Column Data* or *Column Title* to style them separately.

If you have already styled either the title or data for a column and wish to quickly apply the same formatting to the other element, click the appropriate button: *Copy Title Style to Data* or *Copy Data Style to Title*.

3. Under Column Layout, you can change column width and justification.

Width options are:

Maximum

Sets the width according to the length defined in the field format.

Minimum

Sets the width of the column according to the widest value or heading in the field. This is selected by default.

Truncate

Enables you to specify where to truncate the column width based on the specified units (inches or centimeters). To indicate that a field value has been truncated in the browser, WebFOCUS places an exclamation point (!) after every alphanumeric and text field value and an asterisk (*) after every numeric field value.

Wrap

Enables you to specify where to wrap data based on the specified units (inches or centimeters).

Justification options are: left, right, center, or return to the default positioning.

4. Under Graphical options, select the font characteristics, border or grid characteristics, and/or background colors that you wish to apply to the column:

- a. **For borders**, click the *Select Borders* button. The Borders dialog box opens. Select width, style, and/or color options from the drop-down lists.

You can apply the same specifications to all border lines or vary specifications for top, bottom, right, and/or left borders.

Note: To set borders in an HTML report, you must turn Cascading Style Sheets On. Click the *Report Options* icon on the toolbar and select the Output tab. Under *Display Options*, select HTML and verify that On is selected from the Cascading Style Sheets drop-down menu.

When Borders is selected, Grids is disabled.

- b. **For grids**, click the *Select Grids* button. Select a line style and indicate whether to display horizontal lines, vertical lines, or both. This option applies to columns in PDF reports; it does not apply to columns in HTML reports.

When Grids is selected, Borders is disabled.

- c. **For fonts**, click the *Select Fonts* button. The Fonts dialog box opens. Select font name, font style, font size, and/or color.

- d. **For background color**, click the *Single Color* radio button under Background Coloring, and choose a color from the palette.

If you identify the active object as column data, the *Alternating Background Colors* button is activated. You can use this feature to assign colors to alternating rows in one or more columns.

Click *OK* to return to the Style tab.

5. Under *Applying to condition* in the Style tab, you can define or edit a condition that controls when specified formatting options are applied to one or more columns.
6. Click *OK* to return to the matrix where many styling changes will be reflected.

7. Click the *Run* icon on the toolbar to see the column formatting options applied in the report output.

Tip: If you wish to affect other column features, click the:

- **Drill Down tab** to drill- down to another procedure, a URL, or another supported option.
- **General tab** to change the column title or field format, to make the column visible or invisible in the output, or to activate other features.

For details about defining conditional report styles and other column formatting features, see Chapter 3, *Creating Reports With Report Painter*, in the *Creating Reports With Graphical Tools* manual.

Example **Formatting Columns With Data Visualization Graphs and Conditional Styling**

This example uses the joined data source files, CENTGL and CENSYSF (see *Sample Data for FML Hierarchy Examples* on page 2-44) and takes as its starting point the request created in *Displaying One Summary Line for an FML Hierarchy* on page 2-32. However, instead of applying a pre-defined StyleSheet, as you did in the earlier example, you will be formatting the report yourself by applying boldface to each column title, data visualization bar graphs to a numeric column to help you quickly visualize trends and relationships in your data, and conditional styling to the data in a second numeric column.








1. If you completed the referenced example, reopen the Design matrix now. Click the *Report Option* button on the toolbar and in the Style tab, click the *Style File Selection* button. In the dialog box that opens, select *Default* from the Style Sheet Source drop-down list.

If you did not complete the referenced example, follow those instructions up to step 13, then return here.

2. Click the *Report Option* button on the toolbar and select the Output tab. Make sure that *HTML* is the selected Display format and that *Cascading Style Sheets* is turned on. You will need this feature of HTML to display the data visualization graphs. Click OK to return to the matrix.
3. Right-click the Month Actual column and select *Options* from the menu. The Field Properties dialog box opens at the Style tab.
4. Make *Column Title* the active object.
5. Click the *Select Font* button and choose *Bold* from the Font Style box.
6. Click OK. The column title, Month Actual, becomes bold in the matrix.
7. Repeat steps 3-6 for YTD Actual, then click the Field Properties General tab.
8. Click the *Visualize* button. The Data Visualization dialog box opens.

9. Click the *Visualize* check box and select *RED* from the Color drop-down list.
10. Click *OK*. YTD Actual becomes bold in the matrix. You won't see the affect of the data visualization graphics until you run the report.
11. Click the *Run* icon on the toolbar.

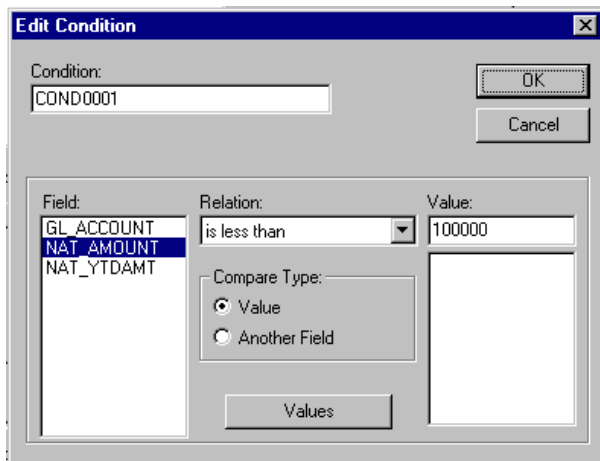
The output is:

| | <u>Month</u>
<u>Actual</u> | <u>YTD</u>
<u>Actual</u> | |
|----------------------|---|---|---|
| Selling Expenses | . | . | |
| Advertising | . | . | |
| TV/Radio | 1,049,146. | 2,954,342. |  |
| Print Media | 244,589. | 721,448. |  |
| Internet Advertising | 9,542. | 29,578. |  |
| Promotional Expenses | 53,719. | 151,732. |  |
| Joint Marketing | 97,135. | 289,799. |  |
| Bonuses/Commissions | 100,188. | 304,199. |  |
| | <hr/> | <hr/> | |
| Selling Expenses | 1,554,319. | 4,451,098. |  |

Suppose that you now want to highlight Monthly Actual Values that are less than 100,00.

12. In the Design matrix, right-click the column title Month Actual and choose Options from the menu. The Field Properties dialog box opens at the Style tab. The active item is *Column Data*. You will now define a condition to control the styling of this data.
13. Click the *Edit Condition* button. The Condition List dialog box opens.
14. Click *New*. The Edit Condition dialog box opens.

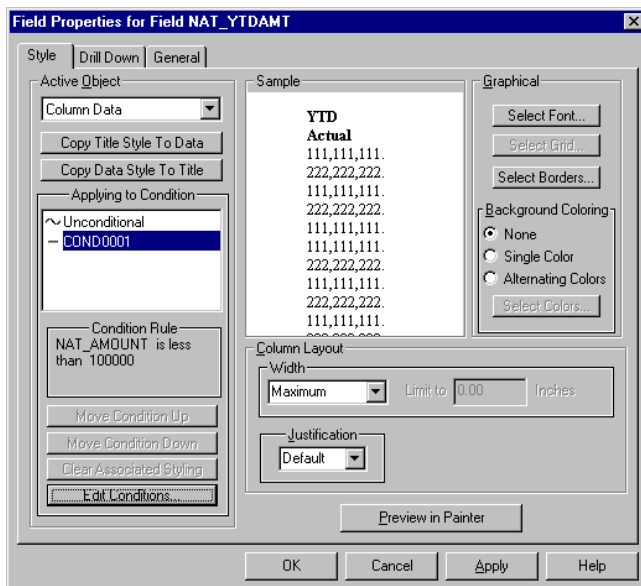
15. Accept the default condition name, COND0001, then select *NAT_AMOUNT* from the Fields box, *is less than* from the Relation drop-down list, and enter *100000* in the Values box.



The **Edit Condition** dialog box is shown. The **Condition:** text box contains "COND0001". The **Field:** list box contains "GL_ACCOUNT", "NAT_AMOUNT" (selected), and "NAT_YTDAMT". The **Relation:** drop-down list shows "is less than". The **Value:** text box contains "100000". The **Compare Type:** section has two radio buttons: "Value" (selected) and "Another Field". There is a "Values" button below the radio buttons. "OK" and "Cancel" buttons are at the top right.

Click **OK**. The Condition List dialog box now shows your entries.

16. If they are correct, click **OK** again to return to the Field Properties dialog box at the Style tab, where you will define the style you want to apply when the condition is met.



The **Field Properties for Field NAT_YTDAMT** dialog box is shown with the **Style** tab selected. The **Active Object** is "Column Data". The **Applying to Condition** list shows "Unconditional" and "COND0001" (selected). The **Condition Rule** section shows "NAT_AMOUNT is less than 100000". The **Sample** area displays a preview of the data with the condition applied. The **Graphical** section includes buttons for "Select Font...", "Select Grid...", "Select Borders...", and "Background Coloring" (with "None" selected). The **Column Layout** section has a "Width" dropdown set to "Maximum" and a "Limit to" value of "0.00" inches. The **Justification** dropdown is set to "Default". There is a "Preview in Painter" button. At the bottom are "OK", "Cancel", "Apply", and "Help" buttons.

17. Click the *Font* button. The Font dialog box opens.

18. Choose *Bold* from the Font Style list and click *OK*. Notice that your selection is reflected in the Sample box.
19. Click the *Run* icon. The values below 100,000 are now displayed in boldface type:

| | <u>Month</u>
<u>Actual</u> | <u>YTD</u>
<u>Actual</u> | |
|----------------------|---|---|--|
| Selling Expenses | . | . | |
| Advertising | . | . | |
| TV/Radio | 1,049,146. | 2,954,342. | |
| Print Media | 244,589. | 721,448. | |
| Internet Advertising | 9,542. | 29,578. | |
| Promotional Expenses | 53,719. | 151,732. | |
| Joint Marketing | 97,135. | 289,799. | |
| Bonuses/Commissions | 100,188. | 304,199. | |
| | <hr/> | <hr/> | |
| Selling Expenses | 1,554,319. | 4,451,098. | |

Procedure How to Format a Row in the Design Grid

1. Right-click a row label (either a default label like R1 or an explicit row label that you have assigned) and select *Options* from the shortcut menu. The Field Properties dialog box opens at the Style tab, with Row identified as the active object.
2. Under *Graphical*, at the right of the Style tab, select the font characteristics, border characteristics, and/or background colors that you wish to apply to the row.

- a. **For borders**, click the *Select Borders* button. The Borders dialog box opens. Select width, style, and/or color options from the drop-down menus.

You can apply the same specifications to all border lines or vary specifications for top, bottom, right, and/or left borders. Click *OK*.

Note: To set borders in an HTML report, you must turn Cascading Style Sheets On. Click the Report Options icon on the toolbar and select the Output tab. Under Display Options select HTML and verify that On is selected from the Cascading Style Sheets drop-down menu.

- b. **For fonts**, click the *Select Fonts* button. The Fonts dialog box opens. Select font name, font style, font size, and color. Click *OK*.
- c. **For background color**, click the *Single Color* radio button under Background Coloring and choose a color from the palette. Click *OK*.

- 3. Click *OK* to return to the Design matrix where many styling changes will be reflected.
- 4. Click the *Run* icon on the toolbar to see the row formatting options applied to the report output.

Example Emphasizing a Row Using Border Lines

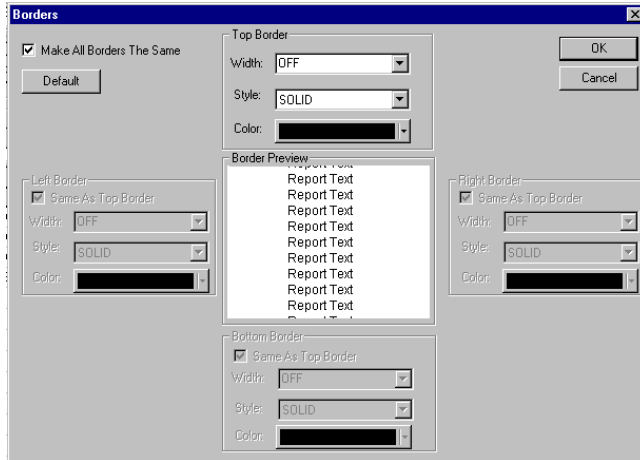
The following example places a thick dashed border around the RECAP row identified by the label TOTCASH.

- 1. Create the report using the sample data source Ledger. In the Report Painter, designate Account as the For field and Amount as a Sum field.
- 2. Click the *Matrix* tab.
- 3. Above the matrix, select the *Use Multiple Values* check box to provide optimal flexibility in reusing tag values (although in this illustration you will use each value only once).
- 4. Drag the tag values 1010, 1020, and 1030 onto the matrix. Right-click each value and choose Row Properties. The TAG dialog box opens. Click the General tab and assign the labels CASH, DD, and TD to rows R1, R2, and R3, respectively. Click *OK* after each entry.
- 5. Create a RECAP row that totals the values of rows CASH (R1), DD (R2), and TD (R3).

One way to quickly create a row for this simple type of calculation is to place your cursor in a column that contains numeric data and highlight the values you want to add. Then click the *Make Recap* icon on the Design toolbar to create the RECAP row. A formula representing the sum of the highlighted values (identified by their row labels) is displayed in the new row, in the column that contained the values. For information about creating more complex RECAP expressions, see *Performing Inter-Row Calculations* on page 2-47.

| Label | 1 | | | | |
|---------|----------|---------|-----------------|--------------|--|
| | Row Type | ACCOUNT | Title | AMOUNT | |
| CASH | TAG | 1010 | CASH ON HAND | 11111 | |
| DD | TAG | 1020 | DEMAND DEPOSITS | 11111 | |
| TD | TAG | 1030 | TIME DEPOSITS | 11111 | |
| TOTCASH | RECAP | | TOTAL CASH | R1 + R2 + R3 | |
| R5 | | | | | |
| R6 | | | | | |
| R7 | | | | | |
| R8 | | | | | |
| R9 | | | | | |
| R10 | | | | | |
| R11 | | | | | |
| R12 | | | | | |
| R13 | | | | | |
| R14 | | | | | |
| R15 | | | | | |
| R16 | | | | | |
| R17 | | | | | |

6. Click the *Report Options* icon on the toolbar above the matrix. On the Style tab, click the *Style File Selection* button (bottom right of window). The Style File Selection dialog box opens. Select the style sheet *DefaultNoGrid* under Style Sheet Source. This will add some basic styling to your report. Click *OK* to confirm your choice and *OK* again to return to the matrix.
7. Right-click the row label *TOTCASH* and select *Options* from the shortcut menu.
8. Click the *Select Borders* button under Graphical options. The Borders dialog box opens.



Note: To set borders in an HTML report, you must turn Cascading Style Sheets On. Click the *Report Options* icon on the toolbar and select the Output tab. Under Display Options select *HTML* and verify that *On* is selected from the Cascading Style Sheets drop-down list.

9. To apply the same specification to the top, bottom, right, and left borders, leave the *Make All Borders the Same* check box selected.
10. In the Top Border box,
 - a. Select *Medium* from the Style drop-down list.
 - b. Select *Dashed* from the Style drop-down list.
11. Click *OK* to return to the Style tab and *OK* again to return to the Design matrix.

12. Click the *Run* icon.

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | 21,239 |

Notice that the dashed border has been placed around the entire row. The gray background is applied by a default style template that you selected in step 6.

In the following variation, specifications are applied separately to the top/bottom and the left/right border lines. A heavy black border line is placed above and below the RECAP row and a thinner dotted line is placed to the left and right of each column in the row.

1. Once again, right-click the row label TOTCASH and select Options from the shortcut menu.
2. Click the *Select Borders* button under Graphical options. The Borders dialog box opens.
3. This time, deselect the *Make All Borders the Same* check box. In the Top Border box:
 - a. Select *Heavy* from the Width drop-down menu.
 - b. Select *Solid* from the Style drop-down menu.
4. In the Bottom border box, click the *Same as Top Border* check box.
5. Next, in both the Left and Right Border boxes, specify Width as *Medium* and Style as *Solid*.

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | 21,239 |

Example Applying Boldface to a Free Text Row

1. Create the report using the sample data source Ledger. In the Report Painter, designate Account as the For field and Amount as a Sum field.
2. Click the *Matrix* tab.
3. In row R1, right-click in the Row Type column, choose *Change type to* from the shortcut menu, and select *Text*. The TEXT dialog box opens.
4. Type ---CASH ACCOUNTS-- in the input box, then right-click in the Label column of row R1 and select *Options*. The Field Properties dialog box opens at the Style tab.
5. Under Graphical, click the *Fonts* button. The Fonts dialog box opens. Select *Bold* in the Font Style list and click *OK*. Click *OK* again to return to the matrix.
6. Drag the tag values 1010, 1020, and 1030 onto the matrix.
7. Right-click in the Row Type column for tag 1010 and select Row Properties. When the TAG dialog box opens, click the General tab and enter the title CASH ON HAND in the Title input box.
8. Repeat steps 6 and 7 for tags 1020 and 1030, using the titles DEMAND DEPOSITS and TIME DEPOSITS, respectively.
9. In row R5, right-click in the Row Type column, choose *Change type to* from the shortcut menu, and select *Text* again. This time leave the input area in the Text dialog box blank to create an empty "text" row. It will appear as a skipped line in the report output.

10. In row R6, create a third text row and type ---OTHER CURRENT ASSETS--- in the input area of the Text dialog box, then right-click in the Label column of row R1 and select *Options*. The Field Properties dialog box opens at the Style tab.
11. To apply boldface to this text row, follow the instructions in step 5.
12. To complete the report, drag the tag values 1100 and 1200 into rows R7 and R8 on the matrix and enter the following row titles for tags 1100 and 1200, ACCOUNTS RECEIVABLE and INVENTORY, respectively, in the TAG dialog box.

The matrix should look as follows:

| Label | | | | 1 | 2 |
|-------|----------|---------|----------------------------|--------|---|
| | Row Type | ACCOUNT | Title | AMOUNT | |
| R1 | TEXT | | ---CASH ACCOUNTS--- | | |
| R2 | TAG | 1010 | CASH ON HAND | 11111 | |
| R3 | TAG | 1020 | DEMAND DEPOSITS | 11111 | |
| R4 | TAG | 1030 | TIME DEPOSITS | 11111 | |
| R5 | TEXT | | | | |
| R6 | TEXT | | ---OTHER CURRENT ASSETS--- | | |
| R7 | TAG | 1100 | ACCOUNTS RECEIVABLE | 11111 | |
| R8 | TAG | 1200 | INVENTORY | 11111 | |
| R9 | | | | | |
| R10 | | | | | |
| R11 | | | | | |
| R12 | | | | | |
| R13 | | | | | |
| R14 | | | | | |
| R15 | | | | | |
| R16 | | | | | |
| R17 | | | | | |
| R18 | | | | | |

13. Click the *Run* icon.

The output is:

| | <u>AMOUNT</u> |
|----------------------------|---------------|
| ---CASH ACCOUNTS--- | |
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| ---OTHER CURRENT ASSETS--- | |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Note: If you were to look at the source code for this procedure, you would see the two free text rows specified using the following code:

```
TYPE=REPORT,
    LABEL=R1
    STYLE=BOLD, $
TYPE=REPORT,
    LABEL=R6
    STYLE=BOLD, $
```

The Financial Report Painter identifies a free text row with a row label designation.

Procedure How to Format a Cell in the Design Matrix

1. Right-click a cell and choose *Options* from the shortcut menu. The Field Properties dialog box opens at the Style tab, with *Cell* identified as the active object.
2. Under *Graphical*, select the font characteristics, border characteristics, and/or background colors that you wish to apply to the cell:
 - a. **For borders**, click the *Select Borders* button. The Borders dialog box opens. Select width, style, and/or color options from the drop-down menus. Click *OK*.
You can apply the same specifications to all border lines or vary specifications for top, bottom, right, and/or left borders.
 - Note:** To set borders in an HTML report, you must turn Cascading Style Sheets On. Click the *Report Options* icon on the toolbar and select the Output tab. Under Display Options, select *HTML* and verify that *On* is selected from the Cascading Style Sheets drop-down menu.
 - b. **For fonts**, click the *Select Fonts* button under Graphical options. The Fonts dialog box opens. Select font name, font style, font size, and color. Click *OK*.
 - c. **For background color**, click the *Single Color* radio button under Background Coloring, and choose a color from the palette. Click *OK*.
3. Under *Applying to condition*, you can define or edit a condition that controls when specified formatting options are applied to a cell.
4. Click *OK* to return to the Design matrix, where most formatting changes will be reflected.
5. Click the *Run* icon on the toolbar to see the cell formatting options applied in the report output.

Note: If you wish to drill- down from the selected cell to another procedure or to a URL, click the Drill Down tab. For details, see *Drilling Down From a Column, Cell, or Row* on page 2-82.

Although certain options on the General tab are enabled when *cell* is the active object, these options will be applied to the entire column, not only to the selected cell.

Example **Highlighting a Cell With Border Lines and Boldface Type**

The following example places a solid line of medium thickness around a cell in the RECAP row identified by the label TOTCASH.

1. Create the report using the sample data source Ledger. In the Report Painter, designate Account as the For field and Amount as a Sum field.
2. Click the *Matrix* tab.
3. Above the matrix, click the *For Multiple* button to provide optimal flexibility in using tag values (although in this illustration you will use each value only once).
4. Drag the tag values 1010, 1020, and 1030 onto the matrix. Notice that the tag values appear in the Title column. To provide more meaningful titles, either type the following entries into the cells on the matrix or open the TAG dialog box and enter the new titles in the General tab. Replace 1010 with CASH on HAND; 1020 with DEMAND DEPOSIT, and 1030 with TIME DEPOSIT.
5. In row R4, right-click on the Row Type column, choose *Change Type to> Bar*. The BAR dialog box opens. Click *OK* to confirm the default underline character. The underline is displayed in the Amounts column, below the values to be summed.
6. In row R5, create a RECAP row that totals the values of rows R1, R2, and R3. Click *OK*.

One way to quickly create a row for this simple type of calculation is to place your cursor in a column that contains numeric data and highlight the values you want to add. Then click the Recap icon on the Design toolbar to create the RECAP row. A formula representing the sum of the highlighted values (identified by their row labels) is displayed in the new row, in the column that contained the values. For information about creating more complex RECAP expressions, see *Performing Inter-Row Calculations* on page 2-47.

7. Change the Title for the RECAP row to TOTAL CASH.

The matrix now looks as follows:

FML Report Properties
☒ Use Multiple Values ☐ First Instance

Row Properties
 Properties: ☐ Invisible: ☐ When Exists: Post To:

TIME DEPOSIT

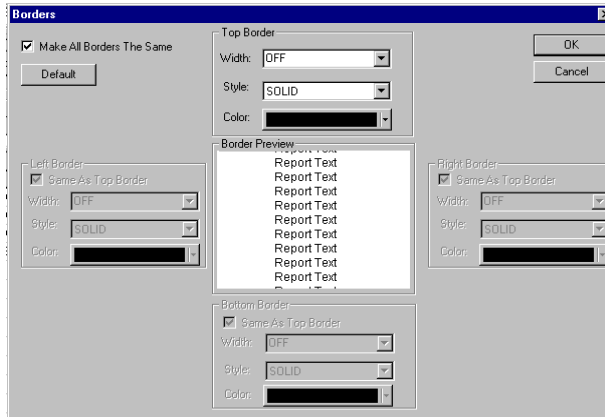
| Label | Row Type | ACCOUNT | Title | 1 | 2 |
|-------|----------|--------------|----------------|----------|---|
| R1 | TAG | 1010 | CASH ON HAND | 11111 | |
| R2 | TAG | 1020 OR 'DEM | DEMAND DEPOSIT | 11111 | |
| R3 | TAG | 1030 | TIME DEPOSIT | 11111 | |
| R4 | BAR | | | ----- | |
| R5 | RECAP | | TOTAL CASH | R1+R2+R3 | |
| R6 | | | | | |
| R7 | | | | | |
| R8 | | | | | |
| R9 | | | | | |
| R10 | | | | | |
| R11 | | | | | |
| R12 | | | | | |
| R13 | | | | | |
| R14 | | | | | |

1010
1020
1030
1100
1200

Report Source Images Matrix

8. Right-click the cell that contains the RECAP formula and select *Options* from the menu. The Field Properties dialog box opens at the Style tab, with Cell identified as the active object.

9. Click the *Select Borders* button under *Graphical* options. The Borders dialog box opens.



Note: To set borders in an HTML report, you must turn Cascading Style Sheets On. Click the *Report Options* icon on the toolbar and select the *Output* tab. Under *Display Format* select *HTML* from the drop-down list and verify that *On* is selected from the *Cascading Style Sheets* drop-down list.

10. To apply the same specification to the top, bottom, right, and left borders, leave the *Make All Borders the Same* check box selected.
11. In the *Top Border* box,
 - a. Select *Medium* from the *Width* drop-down list.
 - b. Select *Solid* from the *Style* drop-down list.
12. Click *OK* to return to the *Style* tab and *OK* again to return to the *Design matrix*.
13. Right-click the same cell and choose *Options* again. This time click the *Select Font* button and choose *Bold* from the *Font Style* list.
14. Click *OK* to return to the *Style* tab and *OK* again to return to the *Design matrix*.
15. Click the *Run* icon.

The output is:

| | AMOUNT |
|----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSIT | 4,494 |
| TIME DEPOSIT | 7,961 |
| TOTAL CASH | 21,239 |

The bold font style and solid border draw attention to the Total in the Amount column.

Procedure How to Apply Conditional Formatting to a Cell

To create a condition:

1. Right-click the cell in the Design matrix and select *Options* from the shortcut menu.
The Field Properties dialog box opens at the Style tab.
2. Click the *Edit Conditions* button.
3. The Condition List dialog box opens.
4. Click *New*.
5. The Edit Condition dialog box opens.
6. Type a name for the condition or accept the default.
7. Select the field for which you want to define a condition from the Field box.
8. Select a relation from the Relations list.
9. Specify a value by doing one of the following: Type a literal value in the Value box.

or

Click the *Values* button to display existing database values and select a value.

or

Click *Another Field* to display other fields in the report and select a field.

10. Click *OK*.

The condition is now added to the Condition List in the Condition List dialog box, from which it can be applied to one or more cells, as well as to columns or rows.

Tip: To create another condition, repeat steps 4 to 10.

11. Click *OK* again to return to the Style tab on the Field Properties dialog box.

The default condition name (e.g., COND0001) appears in the Applying to Condition field box.

The condition you created appears in the Condition Rule box.

12. You are now ready to associate the condition with a style. Select a defined condition from the Applying to Condition dialog box in the Style tab.
13. In the Graphical area of the Style tab, choose one or combination of the following:
 - Click the *Select Font* button to specify conditions such as font, font style and size, color and effects.
 - Click the *Select Grid* button to select a line style (only for PDF reports).
 - Click the *Select Borders* button to select line style, width, and/or color options from the drop-down menus.

Note: To set borders in an HTML report, you must turn Cascading Style Sheets On. Click the *Report Options* icon on the e toolbar and select the Output tab. Under Display Options select *HTML* and verify that *On* is selected from the Cascading Style Sheets drop-down list.
 - In the Background Coloring area, click the *Single Color* radio button to select a background color.

Drilling Down From a Column, Cell, or Row

The drill down feature enables you to add one or more layers of detail to a report by embedding procedures into the report. A drill down “procedure” can be:

- Any type of executable object, such as a report or graph.
- A link to a URL from an HTML or PDF report, or to a JavaScript from an HTML report.
- A link to a Maintain case (function).
- A link to a Maintain procedure.

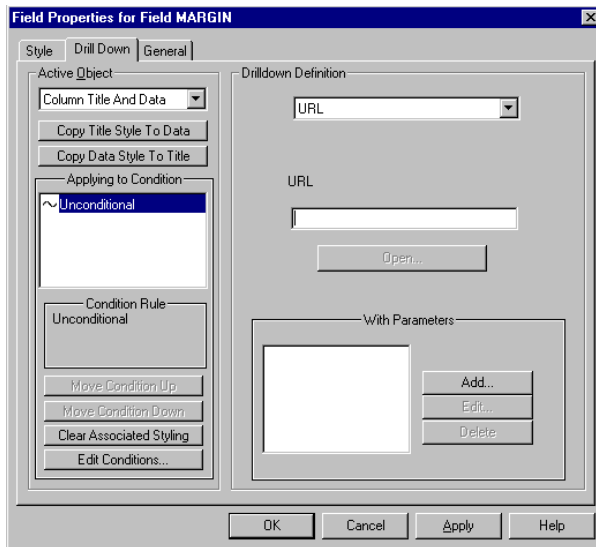
Once the base (summary) report appears in the browser, you can “drill down” to selected data or to a URL or script (or procedure). For example, if you run a report on the country and models of all cars imported into the United States, and you have embedded the appropriate procedures in the report and defined the correct parameters, you can drill down on:

- **ENGLAND**, to view a more detailed report on the sales of all English cars.
- **JAGUAR**, to see specific sales information on the Jaguar.
- A summary report row, to view the values behind each field in the row.

From either the Report Painter or the Financial Report Painter, you can drill down from a column title and its data, or from the title only, or the data only.

In addition, from the Financial Report Painter you can drill down from a single cell in the matrix or from a row or row title.

These behaviors are invoked from the Field Properties Drill Down tab. The active object box indicates the matrix element from which the drill down will be implemented. The Drill-Down Definition indicates what you want to drill down on: a URL, a procedure, or one of the other supported options.



Notice that you can also define conditions that determine when to run an embedded drill down procedure. For example, suppose you have embedded a procedure in the country column of the summary report, but you want your drill down report to appear only for England. You can define a condition that enables the embedded drill down only when country equals England.

The mechanics of establishing a drill-down relationship are the same, regardless of the active object you choose. For details about this process, see Chapter 3, *Creating Reports With Report Painter*, in the *Creating Reports With Graphical Tools* manual.

Procedure How to Access the Drill Down Tab

For a column, right-click the column title,

For a cell, right-click the cell,

then choose *Options* from the shortcut menu. The Field Properties dialog box opens. Click the Drill-Down tab.

- If you select *Options* from a column title, *Column and Title* is the active object. However, you can change the active object to column title or column data.
- If you select *Options* from a cell, *Cell* is the active object. However, you can change the active object to column and title, column title, column data, row title, or row data.

Adding, Inserting, and Deleting Rows

You can add a new row at the bottom of the matrix, insert a row between existing rows, or delete a row from the matrix. The Insert Row and Add Rows dialog boxes provide the additional tabs you need to define a row of the selected type.

***Procedure* How to Add or Insert a Row in the Matrix**

1. Click the *Add* or *Insert* icon on the toolbar.

or

Right-click anywhere in the row and select *Add Row* or *Insert Row* from the shortcut menu.

The Add or Insert Row dialog box opens.

2. Choose a row type: Tag, Recap, Data, Text, Bar or Pickup.

The Options and General tabs adjust to provide the appropriate options for the selected row type. (See topics on specific row types for additional information.)

3. Complete your entries in the Options and General tab.
4. Click *OK* to add the new row to the Row Type column in the matrix.

***Procedure* How to Delete a Row**

Click the *Delete* icon on the toolbar.

or

Right-click anywhere in a row and select *Delete Row* from the shortcut menu.

Editing Row Types and Properties

The Financial Report Painter provides a shortcut menu as an alternative to the toolbar. You can use the shortcut menu to add rows and to change a row's type and properties.

Procedure How to Change Row Types

1. Right-click anywhere in a the row whose type you want to change (except on the row label).
2. Select *Change Type to>* from the shortcut menu.
3. You are asked to verify that you want to change the row type. Click Yes.
4. The Financial Report Painter displays the dialog box you need to complete your row definition. For details, see topics on each row type.

Procedure How to View or Change Row Properties

1. Right-click anywhere in a row (except on the row label).
2. Select *Row Properties* from the shortcut menu.

The dialog box for the selected row type opens.

Procedure How to Cut, Copy, and Paste Rows

To use the standard Windows Clipboard options:

1. Right-click anywhere in the row.
2. Select *Cut Row* or *Copy Row* from the shortcut menu.
3. Right-click a new location on the matrix and choose *Paste Row* from the shortcut menu.

Note: If you Cut and Paste, the row and its label are placed in the new location. If you Copy and Paste, the row contents is copied, but a default label is applied since each row requires a unique label.

Adding and Deleting Columns

Although you initially define the columns in your report in the Report Painter, as you refine your financial report you can include additional columns or delete columns directly from the Financial Report Painter.

Procedure How to Add a Column to a Financial Report

1. Right-click a column title or number and select *Add Column* or *Insert Column* from shortcut menu. The Add Column or Insert Column dialog box opens.
2. Select a field name from the *Select columns to insert* box and click *OK*. The field is either added as the last column in the report or inserted before the column in which you click.

Procedure How to Delete Columns

Right-click a column title or number and select *Delete column* from shortcut menu.

The column is removed from the matrix.

APPENDIX A

Creating Financial Reports With FML Language

Topics:

- Reporting With FML
- Creating Rows From Data
- Supplying Data Directly in a Request
- Performing Inter-Row Calculations
- Referring to Rows in Calculations
- Referring to Columns in Calculations
- Referring to Cells in Calculations
- Using Functions in RECAP Calculations
- Inserting Rows of Free Text
- Adding a Column to an FML Report
- Creating a Recursive Model
- Reporting Dynamically From a Hierarchy
- Customizing a Row Title
- Formatting an FML Report
- Suppressing the Display of Rows
- Saving and Retrieving Intermediate Report Results
- Creating HOLD Files From FML Reports

The Financial Modeling Language (FML) is designed for the special needs associated with creating, calculating, and presenting financially oriented data such as balance sheets, consolidations, or budgets. These reports are distinguished from other reports because calculations are inter-row as well as inter-column and each row or line represents a unique entry or series of entries that can be aggregated directly from the input data or calculated as some function of the data.

Reporting With FML

FML is an integrated extension of the TABLE command. By adding the FOR phrase and the RECAP command, you can handle a vastly expanded range of applications.

Used in conjunction with Dialogue Manager, FML can be used to evaluate “what if” scenarios and develop complete decision support systems. These systems can take advantage of business intelligence features, such as statistical analysis and graphics, in addition to standard financial statements.

Procedures using FML are not hard-wired to the data. As in any other report request, these procedures can easily be changed. FML includes the following facilities:

- Row/column formatting: You can easily specify results in a row-by-row, column-by-column fashion (see *Performing Inter-Row Calculations* on page A-18).
- Intermediate results: You can post FML results to an external file and pick them up at a later time for analysis. This is useful when intermediate results are developed and a final procedure consolidates the results later (see *Saving and Retrieving Intermediate Report Results* on page A-79).
- Inline data entry: FML enables you to specify constants from within the procedure in addition to the data values retrieved from your data source (see *Supplying Data Directly in a Request* on page A-16).
- Recursive reporting: You can produce reports where the results from the end of one time period or column become the starting balance in the next. For example, you could use recursive reports to produce a cash flow projection (see *Creating a Recursive Model* on page A-37).
- Dynamic reporting from charts of accounts or a similar hierarchy of information. You can create a report that changes as the organization of information changes, ensuring that you automatically retrieve information that reflects the latest structure and its values. There is no need to alter either the Master File or the report request. For details, see *Reporting Dynamically From a Hierarchy* on page A-39.

Example Sample FML Request

This example produces a simple asset sheet, contrasting the results of two years. It illustrates many key features of the Financial Modeling Language (FML). Numbers to the left of the procedure lines correspond to explanations that follow the request.

```
TABLE FILE FINANCE
HEADING
"COMPARATIVE ASSET SHEET </2"
SUM AMOUNT ACROSS HIGHEST YEAR
WHERE YEAR EQ '1983' OR '1982'
1. FOR ACCOUNT
2. 1000          AS 'UTILITY PLANT'          LABEL    UTP    OVER
2. 1010 TO 1050  AS 'LESS ACCUMULATED DEPRECIATION' LABEL    UTPAD  OVER
3. BAR
4. RECAP UTPNET=UTP-UTPAD; AS 'TOTAL PLANT-NET' OVER
   BAR
   2000 TO 3999  AS 'INVESTMENTS'          LABEL    INV    OVER
5. "CURRENT ASSETS" OVER
   4000          AS 'CASH'                  LABEL    CASH   OVER
   5000 TO 5999  AS 'ACCOUNTS RECEIVABLE-NET' LABEL    ACR    OVER
   6000          AS 'INTEREST RECEIVABLE'   LABEL    ACI    OVER
   6500          AS 'FUEL INVENTORY'        LABEL    FUEL   OVER
   6600          AS 'MATERIALS AND SUPPLIES' LABEL    MAT    OVER
   6900          AS 'OTHER'                 LABEL    MISC   OVER
   BAR
   RECAP TOTCAS=CASH+ACR+ACI+FUEL+MAT+MISC;AS 'TOTAL CURRENT ASSETS' OVER
   BAR
   7000          AS 'DEFERRED DEBITS'       LABEL    DEFDB  OVER
   BAR
6. RECAP TOTAL=UTPNET+INV+TOTCAS+DEFDB; AS 'TOTAL ASSETS' OVER
   BAR AS '='
   FOOTING
   "</2 *** PRELIMINARY ASSET SHEET BASED ON UNAUDITED FIGURES ***"
END
```

1. FOR and OVER are FML phrases that enable you to structure the report on a row-by-row basis.
2. LABEL assigns a variable name to a row item for use in a RECAP calculation.
1000 and 1010 TO 1050 are tags that identify the data values of the FOR field, ACCOUNT in the FINANCE data source. A report row can be associated with a tag that represents a single data value (like 1000), multiple data values, or a range of values (like 1010 TO 1050).
3. BAR enables you to underline a column of numbers before performing a RECAP calculation.

4. The RECAP command creates a new value based on values already identified in the report with LABEL. In this case, the value UTPNET is derived from UTP and UTPAD and is renamed TOTAL PLANT-NET with an AS phrase to provide it with greater meaning on the report.
5. Like underlines, free text can be incorporated at any point in an FML report.
6. Notice that this RECAP command derives a total (TOTAL ASSETS) from values retrieved directly from the data source and from values derived from previous RECAP computations (UTPNET and TOTCAS).

The output is:

COMPARATIVE ASSET SHEET

| | YEAR | |
|-------------------------------|-----------|-----------|
| | 1983 | 1982 |
| UTILITY PLANT | 1,430,903 | 1,294,611 |
| LESS ACCUMULATED DEPRECIATION | 249,504 | 213,225 |
| TOTAL PLANT-NET | 1,181,399 | 1,081,386 |
| INVESTMENTS | 818 | 5,639 |
| CURRENT ASSETS | | |
| CASH | 4,938 | 4,200 |
| ACCOUNTS RECEIVABLE-NET | 28,052 | 23,758 |
| INTEREST RECEIVABLE | 15,945 | 10,206 |
| FUEL INVENTORY | 35,158 | 45,643 |
| MATERIALS AND SUPPLIES | 16,099 | 12,909 |
| OTHER | 1,264 | 1,743 |
| TOTAL CURRENT ASSETS | 101,456 | 98,459 |
| DEFERRED DEBITS | 30,294 | 17,459 |
| TOTAL ASSETS | 1,313,967 | 1,202,943 |

*** PRELIMINARY ASSET SHEET BASED ON UNAUDITED FIGURES ***

Creating Rows From Data

A normal TABLE request sorts the rows of the report according to the BY phrase you use. The data retrieved is either sorted low-to-high or high-to-low, as requested. The rows may be limited by a screening phrase to a specific subset, but:

- They appear in a sort order.
- Rows appear only for values that are retrieved from the file.
- You can only insert free text between the rows when a sort field changes value, such as:
`ON DIVISION SUBFOOT`
- You can only insert calculations between rows when a sort field changes value, such as:
`ON DIVISION RECAP`

In contrast, the FML FOR phrase creates a matrix in which you can structure your report row-by-row. This organization gives you greater control over the data that is incorporated into a report and over its presentation. You can:

- Report on specific data values for a field in a data source and combine particular data values under a common label, for use in calculations.
- Type data directly into the request to supplement data that is retrieved from the data source.
- Include text, underlines, and calculations at points in the report that are not related to sort breaks.
- Perform recursive processing in which the result of an interim calculation is saved and then used as the starting point for a subsequent calculation.
- Suppress the display of rows for which no data is retrieved.
- Identify rows by labels and columns by numbers, addresses, and values so that you can point to the individual cells formed at each intersection (as on a spreadsheet).

Syntax **How to Retrieve FOR Field Values From a Data Source**

The syntax for specifying rows is

```
FOR fieldname [NOPRINT]
value [OR value OR...] [AS 'text'] [LABEL label] OVER
.
.
[value [OR value ...] [AS 'text'] [LABEL label]
END
```

where:

fieldname

Is a field name in the data source.

value

Is the value (also known as a tag value) describing the data that is retrieved for this row of the report.

AS '*text*'

Allows you to assign a name to tag value. This name will replace the tag value in the output. Enclose the text in single quotation marks.

label

Assigns a label to the row for reference in a RECAP expression. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Even if you assign an explicit label, the positional label (R1, R2, etc,) is retained internally.

By default, a tag value for a FOR field (like 1010) may be added only once to the FML matrix. However, if you wish to add the same value of a FOR field to the matrix more than once, you can turn on the FORMULTIPLE parameter (the default setting is OFF). For details, see *How to Use the Same FOR Field Value in Multiple Rows* on page A-12.

Example Creating Rows From Values in a Data Source

Assume you have a simple data source with financial data for each corporate account, as follows:

CHART OF ACCOUNTS

| ACCOUNT | DESCRIPTION |
|---------|---------------------|
| 1010 | CASH ON HAND |
| 1020 | DEMAND DEPOSITS |
| 1030 | TIME DEPOSITS |
| 1100 | ACCOUNTS RECEIVABLE |
| 1200 | INVENTORY |
| . | . |
| . | . |
| . | . |

Using the FOR phrase in FML, you can issue the following TABLE request in which each value of ACCOUNT is represented by a tag (1010, 1020, etc.) and displays as a separate row:

```
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 OVER
1020 OVER
1030 OVER
1100 OVER
1200
END
```

The output is:

| | <u>AMOUNT</u> |
|------|---------------|
| 1010 | 8,784 |
| 1020 | 4,494 |
| 1030 | 7,961 |
| 1100 | 18,829 |
| 1200 | 27,307 |

Creating Rows From Multiple Records

There are different ways to combine multiple values from your data sources into an FML report row. You can use:

- The OR phrase to sum the values of two or more tags in a single expression. See *How to Sum Values in Rows With the OR Phrase* on page A-9.
- The TO phrase to identify a range of tag values on which to report. See *How to Identify a Range of Values With the TO Phrase* on page A-10.
- A mask to specify a group of tag values without having to name each one. See *How to Use Masking Characters to Retrieve Tag Values* on page A-11.

By default, a FOR field value can only be included in a single row of an FML matrix. However, by turning on the FORMULTIPLE parameter, you can include the same data value in multiple rows in the FML matrix. For example, the same value can exist as a solitary value in one row, be part of a range in another row, and be used in a calculation in a third row. For details, see *How to Use the Same FOR Field Value in Multiple Rows* on page A-12.

In addition to these methods, you can extract multiple tags for a row from an external file. See *Using Tags From External Files* on page A-14.

Syntax**How to Sum Values in Rows With the OR Phrase**

To sum the values of two or more tags in a single report row, use the OR phrase in the FOR phrase. The syntax is

```
FOR fieldname
value1 OR value2 [OR valuen...] [AS 'text'] [LABEL label1] [OVER]
.
.
.
```

where:

fieldname

Is a field name in the data source.

value1, value2, valuen

Are the tags values to be retrieved and summed.

AS '*text*'

Allows you to assign a title to the combined tag values. Enclose the text in single quotation marks.

label1

Assigns a label to the row for reference in a RECAP expression. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Even if you assign an explicit label, the positional label (R1, R2, etc,) is retained internally.

Example Summing Values in Rows

The following model sums the values of three tags (1010, 1020, 1030) as CASH.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 OR 1020 OR 1030 AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| CASH | 21,239 |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Syntax How to Identify a Range of Values With the TO Phrase

To sum the values of a range of tags in a single report row, use the TO phrase in the FOR phrase. The syntax is

```
FOR fieldname
value1 TO value2 [AS ' text' ] [LABEL label1] [OVER]
```

where:

fieldname

Is a field name in the data source.

value1

Is the tag value at the lower limit of the range.

TO

Is the required phrase.

value2

Is the tag value at the upper limit of the range.

AS ' *text*'

Allows you to assign a title to the combined tag values. Enclose the text in single quotation marks.

label1

Assigns a label to the row for reference in a RECAP expression. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Even if you assign an explicit label, the positional label (R1, R2, etc,) is retained internally.

Example Identifying a Range of Values

Since CASH accounts in the LEDGER system are identified by the tags 1010, 1020, 1030, you can specify the range 1010 to 1030:

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 TO 1030 AS 'CASH'
END
```

Syntax How to Use Masking Characters to Retrieve Tag Values

If the tag field has a character (alphanumeric) format, you can perform a masked match. Use the dollar sign character (\$) as the mask. For instance,

```
A$$D
```

matches any four-character value beginning with A and ending with D. The two middle places can be any character. This is useful for specifying a whole group of tag values without having to name each one.

Example Using Masking Characters to Match a Group of Tags

In this example the amounts associated with all four-character accounts that begin with 10, expressed with a mask as 10\$\$, are used to produce the CASH row of the report.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| CASH | 21,239 |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Syntax

How to Use the Same FOR Field Value in Multiple Rows

You can use the same value of a FOR field in many separate rows—whether alone, as part of a range, or in a calculation—by including the following syntax before or within an FML request

```
SET FORMULTIPLE={ON|OFF}
```

or

```
ON TABLE SET FORMULTIPLE {ON|OFF}
```

where:

ON

Enables you to reference the same value of a FOR field in more than one row in an FML request.

With FORMULTIPLE set to ON, a value retrieved from the data source is included on every line in the report output for which it matches the tag references.

OFF

Does not allow you to include the same value in multiple rows. OFF is the default.

With FORMULTIPLE set to OFF, multiple tags referenced in any of these ways (OR, TO, *) are evaluated first for an exact reference or for the end points of a range, next for a mask, and finally within a range. For example, if a value is specified as an exact reference and then as part of a range, the exact reference is displayed. Note that the result will be unpredictable if a value fits into more than one row whose tags have the same priority (for example, an exact reference and the end point of a range.)

For related information, see *Reporting Dynamically From a Hierarchy* on page A-39.

Example Referencing the Same Value in More Than One Row

This request retrieves the tag values for accounts 1010, 1020, and 1030 and lists corresponding values individually; it then aggregates the same values and displays the sum as TOTAL CASH. Similarly, the tag values for accounts 1100 and 1200 are displayed as detail items, and then summarized as TOTAL ASSETS.

```

SET FORMULTIPLE = ON
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
1010 OR 1020 OR 1030 AS 'TOTAL CASH' OVER
" " OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY' OVER
BAR OVER
1100 TO 1200 AS 'TOTAL NON-CASH ASSETS'
END

```

The output is:

| | <u>AMOUNT</u> |
|-----------------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| | <hr/> |
| TOTAL CASH | 21,239 |
| | |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |
| | <hr/> |
| TOTAL NON-CASH ASSETS | 46,136 |

Example **Using Tags From External Files**

In this example, the values for a row of the FML report come from an external file called CASHSTUF, which contains the tags:

```
1010  
1020  
1030
```

The following TABLE request uses the tag values from the external file, summing the amounts in accounts 1010, 1020, and 1030 into the CASH row of the FML report:

```
TABLE FILE LEDGER  
SUM AMOUNT FOR ACCOUNT  
(CASHSTUF) AS 'CASH' OVER  
1100 AS 'ACCOUNTS RECEIVABLE'  
END
```

Notice that the file name must be enclosed in parentheses.

Using the BY Phrase in FML Requests

Only one FOR phrase is permitted in a TABLE request. It substitutes in part for a BY phrase, which controls the sort sequence. However, the request can also include up to 32 BY phrases. In general, BY phrases specify the major (outer) sort fields in FML reports, and the FOR phrase specifies the minor (inner) sort field.

Example Combining BY and FOR Phrases in an FML Request

In this example, the report results for ACCOUNT (the inner sort field) are sorted by REGION (the outer sort field):

```
DEFINE FILE REGION
CUR_YR=E_ACTUAL;
LAST_YR=.831*CUR_YR;
REGION/A4=IF E_ACTUAL NE 0 OR E_BUDGET NE 0 THEN 'EAST' ELSE 'WEST';
END

TABLE FILE REGION
HEADING CENTER
"CURRENT ASSETS FOR REGION <REGION>"
" "
SUM CUR_YR LAST_YR
BY REGION NOPRINT
FOR ACCOUNT
10$$ AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY' OVER
BAR OVER
RECAP CUR_ASSET/I5C = R1 + R2 + R3;
END
```

The output is:

| CURRENT ASSETS FOR REGION EAST | | |
|--------------------------------|---------------|----------------|
| | <u>CUR_YR</u> | <u>LAST_YR</u> |
| CASH | 9,511.00 | 7,903.64 |
| ACCOUNTS RECEIVABLE | . | . |
| INVENTORY | . | . |
| | <hr/> | <hr/> |
| CUR_ASSET | 9,511 | 7,903 |

A sort field value can be used in a RECAP command to allow the model to take different actions within each major sort break. For instance, the following calculation would compute a non-zero value only for the EAST region:

```
RECAP X=IF REGION EQ 'EAST' THEN .25*CASH ELSE 0; AS 'AVAILABLE FOR DIVIDENDS'
```

For more information, see *Performing Inter-Row Calculations* on page A-18.

Supplying Data Directly in a Request

In certain cases, you may need to include some additional constants (for example, exchange rates, inflation rates, etc.) in your model. Not all data values for the model have to be retrieved from the data source. Using FML, you can supply data directly in the request.

Syntax **How to Supply Data Directly in a Request**

```
DATA value, [... , value], $ [AS 'text'] [LABEL label] OVER
```

where:

value

Specifies the values that you are supplying. Values in a list must be separated by commas. The list must end with a comma and a dollar sign (,\$).

AS '*text*'

Allows you to assign a title to the data row. Enclose the text in single quotation marks. Without this entry, the row title is blank on the report.

label

Assigns a name to the data row for use in RECAP calculations. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Example Supplying Data Directly in a Request

In this example, two values (.53 and 1.37) are provided for the exchange rates of marks and pounds, respectively:

```
DEFINE FILE LEDGER
MARKS/I5C=AMOUNT;
POUNDS/I5C=3.2*AMOUNT;
END

TABLE FILE LEDGER
SUM MARKS AS 'GERMAN,DIVISION'
POUNDS AS 'ENGLISH,DIVISION'
FOR ACCOUNT
1010 AS 'CASH--LOCAL CURRENCY' LABEL CASH OVER
DATA .53 , 1.37 , $ AS 'EXCHANGE RATE' LABEL EXCH OVER
RECAP US_DOLLARS/I5C= CASH * EXCH;
END
```

The values supplied are taken one column at a time for as many columns as the report originally specified.

The output is:

| | GERMAN
<u>DIVISION</u> | ENGLISH
<u>DIVISION</u> |
|----------------------|---------------------------|----------------------------|
| CASH--LOCAL CURRENCY | 8,784 | 28,108 |
| EXCHANGE RATE | .53 | 1.37 |
| US_DOLLARS | 4,655 | 38,507 |

Performing Inter-Row Calculations

The RECAP command allows you to perform calculations on data in the rows of the report to produce new rows. You must supply the name and format of the value that results from the calculation, and an expression that defines the calculation you wish to perform. Since RECAP calculations are performed among rows, each row in the calculation must be uniquely identified. FML supplies default row labels for this purpose (R1, R2, etc). However, you may assign more meaningful labels. For details, see *Referring to Rows in Calculations* on page A-19.

Syntax

How to Define Inter-Row Calculations

```
RECAP calcname [/ format] = expression; [AS ' text' ]
```

where:

RECAP

Is the command name and is required. It should begin on a line by itself.

calcname

Is the name you assign to the calculated value. The name can be up to 66 characters long, and must start with an alphabetic character. This name also serves as an explicit label. See *Referring to Rows in Calculations* on page A-19.

format

Is the USAGE format of the calculated value. It cannot exceed the column width. The default is the format of the column in which the calculated value will be displayed.

expression

Can be any calculation available with the DEFINE command (including IF... THEN ... ELSE syntax, functions, excluding DECODE and EDIT, and fields in date format). The expression may extend to as many lines as it requires; a semicolon is required at the end of the expression. For related information see *Using Functions in RECAP Calculations* on page A-30 and the *Using Functions* manual.

The expression can include references to specific rows using the default FML positional labels (R1, R2, etc) or it can refer to rows, columns, and cells using a variety of flexible notation techniques. For details see *Referring to Rows in Calculations* on page A-19, *Referring to Columns in Calculations* on page A-22, and *Referring to Cells in Calculations* on page A-28.

AS ' *text*'

Changes the default title of the row. By default, the name of the RECAP value is displayed as the row title in output. The AS phrase replaces the default title. Enclose the text in single quotation marks.

Reference Usage Notes for RECAP

- RECAP expressions refer to other rows in the model using their labels (either explicit or default). Labels referred to in a RECAP expression must also be specified in the report request.
- The format specified for the RECAP result overrides the format of the column. In the following example,

```
RECAP TOTVAL/D6.2S=IF R1 GT R4 THEN R4 ELSE R1;
AS 'REDUCED VALUE'
```

TOTVAL/D6.2S displays the result as six positions with two decimal places (and displays blanks if the value was zero) in each column of the report, regardless of the format of the data in the column. One use of this feature might be to display percentages in a column of whole numbers.

- Subtotals are not supported in FML.
- In environments that support the RETYPE command, note that RETYPE does not recognize labels in FML with field format redefinition.

Referring to Rows in Calculations

FML assigns a default positional label to each TAG, DATA, RECAP, and PICKUP row. These positional labels are automatically prefixed with the letter R, so that the first such row in the model is R1, the second is R2, etc. You can use these labels to refer to rows in RECAP expressions. (Default labels are not assigned to rows that contain underlines, blank lines, or free text, since these row types need not be referenced in expressions.)

When you refer to rows in a RECAP expression, you can:

- Use the positional row label assigned by FML.
- Create an explicit row label of your own.
- Mix positional and explicit row labels.

If you assign an explicit label, the positional label (R1, R2, etc.) is retained internally.

Note that an explicit label is not needed for a RECAP row because the name of the calculated value on the left of the equal sign can be used as a label.

In addition to their role in RECAP calculations, you can use labels to format rows in an FML report. For details, see *Formatting an FML Report* on page A-59.

Syntax How to Assign an Explicit Row Label

```
rowtype [AS 'text'] LABEL label [OVER]
```

where:

rowtype

Can be a TAG, DATA, or PICKUP row.

AS 'text'

Allows you to assign a different name to the row for the report. Enclose the text in single quotation marks.

label

Assigns a label to a row for reference in a RECAP expression or a StyleSheet declaration. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Even if you assign an explicit label, the positional label (R1, R2, etc,) is retained internally.

Example Referring to Default Row Labels in RECAP Expressions

In this example, FML assigns account 1010 the implicit label R1; account 1020, the implicit label R2; and account 1030, the implicit label R3. Since no label is assigned to a BAR row, the RECAP row is assigned the implicit label R4.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      OVER
1020 AS 'DEMAND DEPOSITS'  OVER
1030 AS 'TIME DEPOSITS'    OVER
BAR                        OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | <u>21,239</u> |

Example Referring to Explicit Row Labels in RECAP Expressions

The following request assigns the labels CA, AR, and INV to three tag rows, which are referenced in the RECAP expression.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                LABEL CA      OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR      OVER
1200 AS 'INVENTORY'           LABEL INV     OVER
BAR                             OVER
RECAP CURASST/I5C= CA + AR + INV;
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| CASH | 21,239 |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |
| | <hr/> |
| CURASST | 67,375 |

Note that the RECAP value could subsequently be referred to by the name CURASST, which functions as an explicit label.

Example Using Labels to Repeat Rows

In certain cases, you may wish to repeat an entire row later in your report. For example, the CASH account can appear in the Asset statement and Cash Flow statement of a financial analysis, as shown below:

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"ASSETS"                OVER
10$$ AS 'CASH' LABEL TOTCASH OVER
.
.
"CASH FLOW"              OVER
RECAP SAMECASH/I5C=TOTCASH; AS 'CASH'
END
```

When you refer to the CASH row the second time, you can use a RECAP calculation (with a new name) and refer to the label, either explicitly (TOTCASH) or implicitly (R1), in the row where CASH was first used.

Tip: If you set the FORMULTIPLE parameter ON, you can repeat the row without giving it another name. For related information, see *Creating Rows From Multiple Records* on page A-8.

Referring to Columns in Calculations

An FML report can refer to explicit columns as well as explicit rows. You can refer to columns using:

- Column numbers.
- Contiguous column notation in RECAP expressions—for example (2,5) to represent columns 2 through 5.
- Column addressing.
- A factor to represent every other column, or every third column, etc.
- Column values.

Example Applying Column Declarations in RECAP Expressions

The following request generates an FML matrix with four rows and three columns of data.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT, YEAR'
    LAST_YR AS 'LAST, YEAR'
COMPUTE CHANGE/I5C = CUR_YR - LAST_YR;
FOR ACCOUNT
1010 AS 'CASH ON HAND'                OVER
1020 AS 'DEMAND DEPOSITS'            OVER
1030 AS 'TIME DEPOSITS'              OVER
BAR                                  OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

Both the columns of the report, as well as the cells of the matrix, can be referenced in another FML report. The output is:

| | CURRENT | LAST | |
|-----------------|-------------|-------------|---------------|
| | <u>YEAR</u> | <u>YEAR</u> | <u>CHANGE</u> |
| CASH ON HAND | 8,784 | 7,216 | 1,568 |
| DEMAND DEPOSITS | 4,494 | 3,483 | 1,011 |
| TIME DEPOSITS | 7,961 | 6,499 | 1,462 |
| TOTAL CASH | 21,239 | 17,198 | 4,041 |

For example, you could use the value 6,499 in another FML report by referring to column 2, row 3. See *Referring to Cells in Calculations* on page A-28.

Referring to Column Numbers in Calculations

A calculation may be performed for one column or for a specific set of columns. To identify the columns, you place the column number in parentheses after the label name.

Example Referring to Column Numbers in a RECAP Expression

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT, YEAR'
LAST_YR AS 'LAST, YEAR'
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH' OVER
" " OVER
RECAP GROCASH(2) / F5.2=100*TOTCASH(1) / TOTCASH(2) - 100;
AS 'CASH GROWTH(%)'
END
```

In the second RECAP expression, note that:

- TOTCASH(1) refers to total cash in column 1.
- TOTCASH(2) refers to total cash in column 2.
- The resulting calculation is displayed in column 2 of the row labeled CASH GROWTH(%).

The RECAP value is only calculated for the column specified.

The output is:

| | CURRENT | LAST |
|-----------------|-------------|-------------|
| | <u>YEAR</u> | <u>YEAR</u> |
| CASH ON HAND | 8,784 | 7,216 |
| DEMAND DEPOSITS | 4,494 | 3,483 |
| TIME DEPOSITS | 7,961 | 6,499 |
| TOTAL CASH | 21,239 | 17,198 |
| CASH GROWTH(%) | | 23.50 |

After data retrieval is completed, a single column is calculated all at once, and multiple columns one by one.

Referring to Contiguous Columns in Calculations

When a set of contiguous columns is needed within a RECAP, you can separate the first and last column numbers with commas. For example, DIFFERENCE (2,5) indicates that you want to compute the results for columns 2 through 5.

Example Recapping Over Contiguous Columns

In this example the RECAP calculation for ATOT occurs only for columns 2 and 3, as specified in the request. No calculation is performed for column 1.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END

TABLE FILE LEDGER
SUM NEXT_YR CUR_YR LAST_YR
FOR ACCOUNT
10$$ AS 'CASH'                                OVER
1100 AS 'ACCOUNTS RECEIVABLE'                 OVER
1200 AS 'INVENTORY'                           OVER
BAR                                            OVER
RECAP ATOT(2,3)/I5C = R1 + R2 + R3;
AS 'ASSETS--ACTUAL'
END
```

The output is:

| | <u>NEXT_YR</u> | <u>CUR_YR</u> | <u>LAST_YR</u> |
|---------------------|----------------|---------------|----------------|
| CASH | 25,992 | 21,239 | 17,198 |
| ACCOUNTS RECEIVABLE | 21,941 | 18,829 | 15,954 |
| INVENTORY | 31,522 | 27,307 | 23,329 |
| ASSETS--ACTUAL | | 67,375 | 56,481 |

Referring to Column Addresses in Calculations

When you need a calculation not for every column, but for every other, or every third column, you can supply a factor, or column address, to do this. Column addressing is particularly useful when several data fields are displayed within each value of a column sort.

Syntax How to Use Column Addressing in a RECAP Expression

The left-hand side of the expression has the form

value(*s*, *e*, *i*) [*/ format*] =

where:

value

Is the name you assign to the result of the RECAP calculation.

s

Is the starting column.

e

Is the ending column (may be * to denote all columns).

i

Is the increment factor.

format

Is the USAGE format of the calculated value. The default value is the format of the original column.

Example Applying Column Addressing in a RECAP Expression

In the following statement, there are two columns for each month:

SUM ACTUAL AND FORECAST ACROSS MONTH

If you want to perform a calculation only for the ACTUAL data, you can control the placement of the results with a RECAP in the form:

RECAP *calcname*(1, *, 2) = *expression*;

The asterisk means to continue the RECAP for all odd-numbered columns (beginning in column 1, with an increment of 2, for all columns).

Referring to Relative Column Addresses in Calculations

A calculation can use a specific column as a base and refer to all other columns by their displacement from that column. The column to the left of the base column has a displacement of -1 relative to the base column; the column to the right has a displacement of +1. For example,

```
COMP=FIX(*)-FIX(*-1);
```

can refer to the change in fixed assets from one period to the next. The reference to COMP=FIX(*) is equivalent to COMP=FIX.

When referring to a prior column, it must already have been retrieved or its value is zero.

Example Applying Relative Column Addressing in a RECAP Expression

This example computes the change in cash (CHGCASH) for columns 1 and 2.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END

TABLE FILE LEDGER
SUM NEXT_YR CUR_YR LAST_YR
FOR ACCOUNT
10$$ AS 'TOTAL CASH' LABEL TOTCASH OVER
" " OVER
RECAP CHGCASH(1,2)/I5C = TOTCASH(*) - TOTCASH(*+1); AS 'CHANGE IN CASH'
END
```

The output is:

| | <u>NEXT_YR</u> | <u>CUR_YR</u> | <u>LAST_YR</u> |
|----------------|----------------|---------------|----------------|
| TOTAL CASH | 25,992 | 21,239 | 17,198 |
| CHANGE IN CASH | 4,753 | 4,041 | |

Referring to Column Values in Calculations

When a report is sorted using the ACROSS phrase, all of the retrieved values are aligned under their appropriate columns. Each column has a title consisting of one value of the ACROSS field. The entire column of data can be directly addressed by this value in a RECAP calculation.

Example Referring to a Column by Its Value in a RECAP Expression

The following request uses a factor that depends on the value of the ACROSS field (YEAR) to calculate the inventory cost for each year. It then calculates the profit by summing the assets and subtracting the inventory cost for each year.

```
TABLE FILE LEDGER
SUM AMOUNT ACROSS YEAR
FOR ACCOUNT
10$$ AS 'CASH' LABEL CASH OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL RECEIVE OVER
BAR OVER
1200 AS 'INVENTORY VALUE' LABEL INVENT OVER
RECAP INVENTORY_FACTOR/F5.2 = IF YEAR LT '1986'
THEN 1.1 ELSE 1.25; AS 'INVENTORY COST FACTOR' OVER
RECAP INVENTORY_COST = INVENTORY_FACTOR * INVENT;
AS 'INVENTORY COST' OVER
BAR OVER
RECAP PROFIT = CASH + RECEIVE - INVENTORY_COST;
END
```

The output is:

| | YEAR | | |
|-----------------------|-------|--------|--------|
| | 1985 | 1986 | 1987 |
| CASH | 5,663 | 7,001 | 8,575 |
| ACCOUNTS RECEIVABLE | 5,295 | 6,250 | 7,284 |
| INVENTORY VALUE | 7,754 | 9,076 | 10,477 |
| INVENTORY COST FACTOR | 1.10 | 1.25 | 1.25 |
| INVENTORY COST | 8,529 | 11,345 | 13,096 |
| PROFIT | 2,429 | 1,906 | 2,763 |

Referring to Cells in Calculations

You can refer to columns and rows using a form of cell notation that identifies the intersection of a row and a column as (r, c).

Syntax **How to Use Cell Notation for Rows and Columns in a RECAP Expression**

A row and column can be addressed in an expression by the notation

$E(r, c)$

where:

E

Is a required constant.

r

Is the row number.

c

Is the column number. Use an asterisk (*) to indicate the current column.

Example Referring to Columns Using Cell Notation in a RECAP Expression

In this request, two RECAP expressions derive VARIANCEs (EVAR and WVAR) by subtracting values in four columns (1, 2, 3, 4) in row three (PROFIT); these values are identified using cell notation (r,c).

```
TABLE FILE REGION
SUM E_ACTUAL E_BUDGET W_ACTUAL W_BUDGET
FOR ACCOUNT
3000 AS 'SALES' OVER
3100 AS 'COST' OVER
BAR OVER
RECAP PROFIT/I5C = R1 - R2; OVER
" " OVER
RECAP EVAR(1)/I5C = E(3,1) - E(3,2);
AS 'EAST--VARIANCE' OVER
RECAP WVAR(3)/I5C = E(3,3) - E(3,4);
AS 'WEST--VARIANCE'
END
```

The output is:

| | <u>E ACTUAL</u> | <u>E BUDGET</u> | <u>W ACTUAL</u> | <u>W BUDGET</u> |
|---------------|-----------------|-----------------|-----------------|-----------------|
| SALES | 6,000 | 4,934 | 7,222 | 7,056 |
| COST | 4,650 | 3,760 | 5,697 | 5,410 |
| PROFIT | 1,350 | 1,174 | 1,525 | 1,646 |
| EAST-VARIANCE | 176 | | | |
| WEST-VARIANCE | | | -121 | |

Note: In addition to illustrating cell notation, this example demonstrates the use of column numbering. Notice that the display of the EAST and WEST VARIANCEs in columns 1 and 3, respectively, are controlled by the numbers in parentheses in the request: EVAR (1) and WVAR (3).

Using Functions in RECAP Calculations

You may provide your own calculation routines in RECAP rows to perform special-purpose calculations, a useful feature when these calculations are mathematically complex or require extensive look-up tables.

User-written functions are coded as subroutines in any language that supports a call process, such as FORTRAN, COBOL, PL/1, and BAL. See the *Using Functions* manual for information about creating your own functions.

Syntax

How to Call a Function in a RECAP Command

```
RECAP calcname [ (s,e,i) ] [ / format ] = function (input1, ..., inputn, ' format2' );
```

where:

calcname

Is the name you assign to the calculated value.

(*s,e,i*)

You can specify a start (*s*), end (*e*), and increment (*i*) value for the column where you want the value displayed; if omitted the value appears in all columns.

format

The format for the calculation is optional; the default is the format of the column. If the calculation consists of only the subroutine, make sure that the format of the subroutine output value (*format2*) agrees with the calculation's format. If the calculation format is larger than the column width, the value displays in that column as asterisks.

function

Is the eight-character name of the subroutine. It must be different from any row label and cannot contain any of the following special characters: = - , / ().

input

Are the input arguments for the call to the function; they may include numeric constants, alphanumeric literals, row and column references (R notation, E notation, or labels), or names of other RECAP calculations.

Make sure that the values being passed to the function agree in number and type with the arguments as coded in the function.

format2

Is the format of the return value, which must be enclosed in single quotation marks.

Example Calling a Function in a RECAP Command

Suppose you have a function named INVEST in your private collection of functions (INVEST is not available in the supplied library) and it calculates the amount on the basis of cash on hand, total assets, and the current date. In order to create a report that prints an account of company assets and calculates how much money the company has available to invest, you must create a report request that invokes the INVEST function.

The current date is obtained from the &YMD system variable. The NOPRINT option beside it prevents the date from appearing in the report; the date is solely used as input for the next RECAP statement.

The request is:

```
TABLE FILE LEDGER
HEADING CENTER
"ASSETS AND MONEY AVAILABLE FOR INVESTMENT </2"
SUM AMOUNT ACROSS HIGHEST YEAR
IF YEAR EQ 1985 OR 1986
FOR ACCOUNT
1010 AS 'CASH'                                LABEL CASH      OVER
1020 AS 'ACCOUNTS RECEIVABLE'                 LABEL ACR        OVER
1030 AS 'INTEREST RECEIVABLE'                 LABEL ACI        OVER
1100 AS 'FUEL INVENTORY'                     LABEL FUEL       OVER
1200 AS 'MATERIALS AND SUPPLIES'              LABEL MAT        OVER
BAR                                             OVER
RECAP TOTCAS = CASH+ACR+ACI+FUEL+MAT; AS 'TOTAL ASSETS' OVER
BAR                                             OVER
RECAP THISDATE/A8 = &YMD; NOPRINT              OVER
RECAP INVAIL = INVEST(CASH,TOTCAS,THISDATE,'D12.2'); AS
      'AVAIL. FOR INVESTMENT'                  OVER
BAR AS '='
END
```

The output is:

| ASSETS AND MONEY AVAILABLE FOR INVESTMENT | | |
|---|--------|--------|
| | YEAR | |
| | 1986 | 1985 |
| CASH | 2,100 | 1,684 |
| ACCOUNTS RECEIVABLE | 875 | 619 |
| INTEREST RECEIVABLE | 4,026 | 3,360 |
| FUEL INVENTORY | 6,250 | 5,295 |
| MATERIALS AND SUPPLIES | 9,076 | 7,754 |
| TOTAL ASSETS | 22,327 | 18,712 |
| AVAIL. FOR INVESTMENT | 3,481 | 2,994 |

Inserting Rows of Free Text

You can insert text anywhere in your FML report by typing it on a line by itself and enclosing it within double quotation marks. You can also add blank lines, designated as text, to improve the appearance of the report.

In addition, you can include data developed in your FML report in a row of free text by including the label for the data variable in the text row.

Example Inserting Free Text

In this example, three rows of free text are inserted, one blank and two text rows:

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---"          OVER
1010 AS 'CASH ON HAND'            OVER
1020 AS 'DEMAND DEPOSITS'         OVER
1030 AS 'TIME DEPOSITS'           OVER
" "                               OVER
" --- OTHER CURRENT ASSETS ---"   OVER
1100 AS 'ACCOUNTS RECEIVABLE'     OVER
1200 AS 'INVENTORY'
END
```

The output is:

| | <u>AMOUNT</u> |
|------------------------------|---------------|
| --- CASH ACCOUNTS --- | |
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| | |
| --- OTHER CURRENT ASSETS --- | |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Notice that the blank row was created by enclosing a blank within double quotation marks on a separate line of the report request.

Syntax

How to Insert Data Variables In Text Rows

" *text* < *label* [(*c*)] [>] "

where:

<

Is a required left caret to bracket the label.

label

Is the explicit or implicit row label. (In a RECAP, the calculated value functions as the label.)

c

Is an optional cell identifier that indicates the column number of the cell. This identifier, however, is required whenever there is more than one column in the report. If you use it, enclose it in parentheses.

>

Is an optional right bracket that can be used to make the positioning clearer.

Example Inserting a Data Variable in a Text Row

In this example, the RECAP value CURASST is suppressed by the NOPRINT command and inserted instead as a data variable in the text row.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'Cash'                                LABEL CA      OVER
1100 AS 'Accounts Receivable'                 LABEL AR      OVER
1200 AS 'Inventory'                           LABEL INV     OVER
RECAP CURASST/I5C= CA + AR + INV; NOPRINT OVER
"Current Assets: <CURASST"
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| Cash | 21,239 |
| Accounts Receivable | 18,829 |
| Inventory | 27,307 |
| Current Assets: | 67,375 |

Adding a Column to an FML Report

The request controls the number of columns in any report. For instance, if a request contains the display command SUM AMOUNT AND FORECAST, the report will contain two columns: AMOUNT and FORECAST.

You can add columns in an FML request, just as you can in a TABLE request, using the COMPUTE command to calculate a value or simply to allocate the space, column title, and format for a column. For related information see Chapter 5, *Creating Temporary Fields* in the *Creating Reports With WebFOCUS Language* manual.

Example Adding a Column to an FML Report

This example uses a COMPUTE command to generate the calculated value CHANGE and display it as a new column in the FML report. The following request generates an FML matrix with four rows and three columns of data.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT, YEAR'
    LAST_YR AS 'LAST, YEAR'
COMPUTE CHANGE/I5C = CUR_YR - LAST_YR;
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

The output is:

| | CURRENT | LAST | |
|-----------------|-------------|-------------|---------------|
| | <u>YEAR</u> | <u>YEAR</u> | <u>CHANGE</u> |
| CASH ON HAND | 8,784 | 7,216 | 1,568 |
| DEMAND DEPOSITS | 4,494 | 3,483 | 1,011 |
| TIME DEPOSITS | 7,961 | 6,499 | 1,462 |
| TOTAL CASH | 21,239 | 17,198 | 4,041 |

Note: The designated calculation is performed on each tag or RECAP row of the report. The RECAP rows, however, may change the calculation.

Example Adding a New Time Period as a Column

The following request adds a future time period to a report:

```
DEFINE FILE LEDGER
CUR_YR/P5C = AMOUNT;
LAST_YR/P5C = .87*AMOUNT - 142;
END

TABLE FILE LEDGER
SUM AMOUNT
ACROSS YEAR AND COMPUTE 1999/P5C = 2.5*AMOUNT;
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER

RECAP TOTCASH/P5C = R1 + R2 + R3; AS 'TOTAL CASH' OVER
RECAP CHANGE(2,*) = TOTCASH(*) - TOTCASH(*-1);
END
```

The output is:

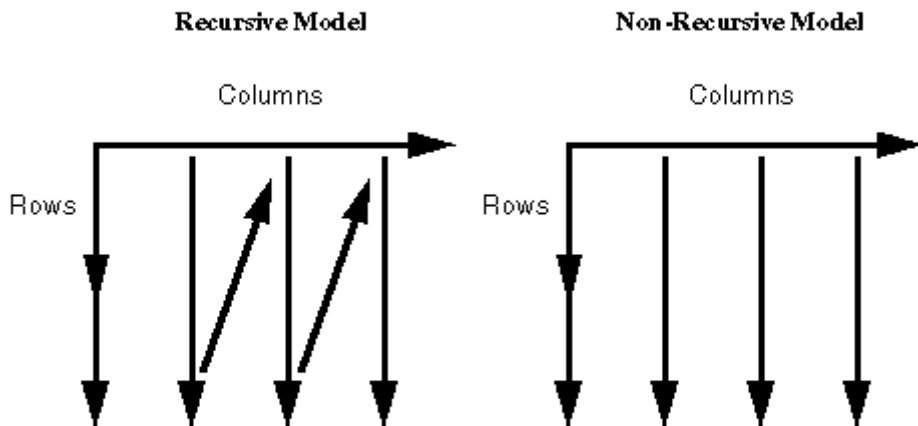
| | YEAR | | | |
|-----------------|-------|-------|-------|--------|
| | 1985 | 1986 | 1987 | 1999 |
| CASH ON HAND | 1,684 | 2,100 | 5,000 | 4,210 |
| DEMAND DEPOSITS | 619 | 875 | 3,000 | 1,548 |
| TIME DEPOSITS | 3,360 | 4,026 | 575 | 8,400 |
| TOTAL CASH | 5,663 | 7,001 | 8,575 | 14,158 |
| CHANGE | | 1,338 | 1,574 | 5,583 |

Creating a Recursive Model

Models involving different time periods often require using the ending value of one time period as the starting value for the next time period. The series of calculations describing these situations has two characteristics:

- The labels on one or more RECAP rows are duplicates of other rows. That is, they are used repeatedly to recompute some values.
- A calculation may refer to a label not yet described, but provided later in the model. If, at the end of the model, a label that is needed is missing, an error message is displayed.

Recursive models require that the columns are produced in sequential order, one by one. In nonrecursive models, all of the columns can be produced simultaneously. Schematically, these patterns are shown below.



FML automatically switches to sequential order as soon as either of the two modeling conditions requiring the switch is recognized (that is, either reuse of labels by different rows, or forward reference to a label in a calculation).

Example Creating a Recursive Model

The following example illustrates recursive models. Note that one year’s ENDCASH becomes the next year’s STARTING CASH.

```
DEFINE FILE REGION
CUR_YR=E_ACTUAL;
LAST_YR=.831*CUR_YR;
NEXT_YR=1.2297*CUR_YR;
END

TABLE FILE REGION
SUM LAST_YR CUR_YR NEXT_YR
FOR ACCOUNT
10$$ AS 'STARTING CASH' LABEL STCASH          OVER
RECAP STCASH(2,*) = ENDCASH(*-1);              OVER
" "                                           OVER
3000 AS 'SALES' LABEL SLS                      OVER
3100 AS 'COST' LABEL COST                     OVER
BAR                                           OVER
RECAP PROFIT/I5C = SLS - COST;                 OVER
" "                                           OVER
RECAP ENDCASH/I5C = STCASH + PROFIT;
END
```

The output is:

| | <u>LAST_YR</u> | <u>CUR_YR</u> | <u>NEXT_YR</u> |
|---------------|----------------|---------------|----------------|
| STARTING CASH | 7,903.64 | 9,024.00 | 10,374.00 |
| SALES | 4,986.00 | 6,000.00 | 7,378.20 |
| COST | 3,864.15 | 4,650.00 | 5,718.11 |
| PROFIT | <u>1,121</u> | <u>1,350</u> | <u>1,660</u> |
| ENDCASH | 9,024 | 10,374 | 12,034 |

Reporting Dynamically From a Hierarchy

Hierarchical relationships between fields can be defined in a Master File and automatically displayed using the Financial Modeling Language (FML). The parent and child fields must share data values and their relationship should be hierarchical. The formats of the parent and child fields must both be numeric or both alphanumeric.

For example, suppose that:

- An employee data source contains both the employee's ID and the ID of the employee's manager.
- or
- A general ledger data source contains both an account number field and an account parent field.

By examining these fields, it is possible to construct the entire organization chart or charts of accounts structure. However, to print the chart in a traditional FML report, you would have to list the employee IDs or account numbers in the request syntax in the order in which they should appear on the report. If an employee or account is added, removed, or transferred, you would have to change the report request to reflect this change in organizational structure. For example:

```
TABLE FILE EMPLOYEE
PRINT DEPARTMENT CURR_JOBCODE
FOR EMP_ID
999999999 OVER
222222222 OVER
.
.
.
```

In contrast, with FML hierarchies you can define the hierarchical relationship between two fields in the Master File and load this information into memory. The FML request can then *dynamically* construct the rows that represent this relationship and display them in the report starting at any point in the hierarchy. In the example shown, EMP_ID is called the hierarchy field.

Requirements for FML Hierarchies

1. In the Master File, use the `PROPERTY=PARENT_OF` and `REFERENCE=hierarchyfld` attributes to define the hierarchical relationship between two fields.
2. The hierarchy must be loaded into memory. The loaded hierarchy is called a chart. If the hierarchy is defined in the Master File and referenced by the FML request, the hierarchy is loaded automatically. If you want to use a hierarchy defined in a Master File that will not be either referenced in the FML request or joined to the Master File referenced in the FML request, you must issue the `LOAD CHART` command before issuing the FML request.

The number of charts that can be loaded is 16. Charts are automatically unloaded when the session ends.

3. In the `FOR` phrase of the FML request, use the `GET/WITH CHILDREN` or `ADD` phrase to retrieve the hierarchical data starting at a specific point in the hierarchy.

To use FML hierarchies, the `FOR` field must either be:

- The hierarchy field.
or
- Used as the join field to a unique segment that has the hierarchy field. In this case the hierarchy field must be the join field. Note that the condition that the join be unique only applies if the hierarchy is defined in the cross-referenced segment.

In other words, the `FOR` field must be in a parent-child hierarchy or it must be linked to a parent-child hierarchy. The latter case allows transaction data that contains the hierarchy field to be joined to a separate data source that contains the hierarchy definition.

As with any FML request, a tagged row is displayed even if no data is found in the file for the tag values, with a period (.) representing the missing data. You can override this convention by adding the phrase `WHEN EXISTS` to the definition of a tagged row. This makes displaying a row dependent upon the existence of data for the tag.

Example Defining a Hierarchy in a Master File

The CENTGL Master File contains a charts of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field:

```
FILE=CENTGL          , SUFFIX=FOC
SEGNAME=ACCOUNTS, SEGTYPE=S01
FIELDNAME=GL_ACCOUNT,          ALIAS=GLACCT,  FORMAT=A7,
                                TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT,    ALIAS=GLPAR,   FORMAT=A7,
                                TITLE=Parent,
                                PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE,      ALIAS=GLTYPE,  FORMAT=A1,
                                TITLE=Type, $
FIELDNAME=GL_ROLLUP_OP,         ALIAS=GLROLL,  FORMAT=A1,
                                TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL,     ALIAS=GLLEVEL, FORMAT=I3,
                                TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION,    ALIAS=GLCAP,   FORMAT=A30,
                                TITLE=Caption,
                                PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT,          ALIAS=ALINE,   FORMAT=A6,
                                TITLE='System,Account,Line', MISSING=ON, $
```

The CENTSYSF data source contains detail-level financial data. This is unconsolidated financial data for a fictional corporation, CenturyCorp. It is designed to be separate from the CENTGL database as if it came from an external accounting system. It uses a different account line system (SYS_ACCOUNT) which can be joined to the SYS_ACCOUNT field in CENTGL. Data uses "natural" signs (expenses are positive, revenue negative).

```
FILE=CENTSYSF        , SUFFIX=FOC
SEGNAME=RAWDATA      , SEGTYPE=S2
FIELDNAME = SYS_ACCOUNT , , A6 , FIELDT
PE=I,
                                TITLE='System,Account,Line', $
FIELDNAME = PERIOD      , , YYM , FIELDTYPE=I, $
FIELDNAME = NAT_AMOUNT  , , D10.0 , TITLE='Month,Actual', $
FIELDNAME = NAT_BUDGET  , , D10.0 , TITLE='Month,Budget', $
FIELDNAME = NAT_YTDAMT  , , D12.0 , TITLE='YTD,Actual', $
```

Displaying an FML Hierarchy

The GET CHILDREN and WITH CHILDREN commands dynamically retrieve and display hierarchical data on the FML report. GET CHILDREN displays only the children, not the parent value referenced in the command. WITH CHILDREN displays the parent and then the children.

Syntax

How to Display an FML Hierarchy

```
TABLE FILE filename
{PRINT|SUM} ....
FOR hierarchyfld
parentvalue {GET|WITH} CHILD[REN] [n|ALL] [AS CAPTION| 'text' ] [LABEL label]
.
.
.
END
```

where:

filename

Is the name of the file to be used in the FML request. (If the hierarchy for this request cannot be loaded automatically, you must have previously issued the LOAD CHART command to load the hierarchy.)

hierarchyfld

Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

parentvalue

Is the parent value for which the children are to be retrieved.

GET CHILDREN

Displays the hierarchy starting from the first child of the specified *parentvalue*. Does not include the parent in the display. (This corresponds to the FML syntax CHILD1 OVER CHILD2 OVER ...)

WITH CHILDREN

Displays the hierarchy starting from the specified *parentvalue*. Includes the parent in the display. (This corresponds to the FML syntax *parentvalue* OVER CHILD1 OVER CHILD2 OVER ...)

`n/ALL`

Is a positive integer from 1 to 99 specifying the number of levels of the hierarchy to display. If a number greater than 99 is specified, a warning message is displayed and *n* is set to 99. The default value is 1. Therefore, if *n* is omitted, direct children are displayed. GET or WITH CHILDREN 2 displays direct children and grandchildren. GET or WITH CHILDREN 99 displays children to 99 levels. ALL is a synonym for 99. Each child instance is printed over the next child instance. Successive levels of the hierarchy field are indented two spaces from the previous level.

`CAPTION`

Indicates that the caption values to display should be taken from the field defined as the CAPTION in the Master File.

Note that the AS CAPTION phrase is supported for tagged rows, including those that do not use the GET/WITH CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

`'text'`

Is a text string to use as the row title for the hierarchy field values. The CAPTION field defined in the Master File will not be used as the caption on the report output.

`label`

Is an explicit row label. Each generated row is labeled with the specified label text.

Note: The hierarchy is displayed sorted by the parent field and, within parent, sorted by the hierarchy field.

Example Displaying an FML Hierarchy

The following request displays two levels of account numbers, starting from account 3000:

```
SET BLANKINDENT=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
3000 WITH CHILDREN 2
END
```

The output is:

| | <u>Parent</u> |
|------|---------------|
| 3000 | 1000 |
| 3100 | 3000 |
| 3110 | 3100 |
| 3120 | 3100 |
| 3130 | 3100 |
| 3140 | 3100 |
| 3200 | 3000 |
| 3300 | 3200 |
| 3400 | 3200 |
| 3500 | 3200 |
| 3600 | 3200 |
| 3700 | 3200 |
| 3800 | 3200 |
| 3900 | 3200 |

Note that if the request had specified GET CHILDREN instead of WITH CHILDREN, the line for the parent value (3000) would not have displayed on the report output.

Example Displaying an FML Hierarchy With Captions

The following request displays two levels of a charts of accounts hierarchy, starting with account 1000 (the top of the hierarchy), and displays the caption field values instead of the account numbers:

```
SET BLANKINDENT=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
1000 WITH CHILDREN 2 AS CAPTION
END
```

The output is:

| | <u>Parent</u> |
|--------------------------|---------------|
| Profit Before Tax | |
| Gross Margin | 1000 |
| Sales Revenue | 2000 |
| Cost Of Goods Sold | 2000 |
| Total Operating Expenses | 1000 |
| Selling Expenses | 3000 |
| General + Admin Expenses | 3000 |
| Total R+D Costs | 1000 |
| Salaries | 5000 |
| Misc. Equipment | 5000 |

Note that if the request had specified GET CHILDREN instead of WITH CHILDREN, the line for the parent value (1000, Profit Before Tax) would not have displayed on the report output.

Consolidating an FML Hierarchy

The ADD command consolidates multiple levels of the hierarchy on one line of the FML report output. ADD can be used alone or in conjunction with GET CHILDREN or WITH CHILDREN. Note that ADD is designed to work with requests that use the SUM command. It is also designed to be used with detail level data, not data that is consolidated.

When used alone, ADD aggregates the parent and children on one line of the report output, summing the numeric data values included on the line. (This corresponds to the FML syntax parentvalue or CHILD1 OR CHILD2 OR ...)

When used in conjunction with GET CHILDREN, ADD displays one line for each child of the specified parent value. Each line is a summation of that child and all of its children. You can specify the number of levels of children to display (which determines the number of lines generated on the report output) and the depth of summation under each child. By default, only direct children will have a line in the report output and the summary for each child will include all of its children.

When used in conjunction with WITH CHILDREN, ADD first displays a line in the report output that consists of the summation of the parent value and all of its children. Then it displays additional lines identical to those displayed by GET CHILDREN ADD.

In order to use a data record in more than one line of an FML report (for example, to display both detail and summary lines or to consolidate detail data at multiple levels), the following setting is required:

`SET FORMULTIPLE=ON`

Syntax**How to Create One Summary Row for an FML Hierarchy**

```
TABLE FILE filename
SUM....
FOR hierarchyfld
parentvalue ADD [n|ALL] [AS CAPTION|'text'] [LABEL label]
.
.
.
END
```

where:

filename

Is the name of the file to be used in the FML request. (If the hierarchy for this request cannot be loaded automatically, you must have previously issued the LOAD CHART command to load the hierarchy.)

hierarchyfld

Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

parentvalue

Is the parent value that determines the starting point in the hierarchy for the aggregation.

n|ALL

Is a positive integer from 1 to 99 specifying the number of levels of the hierarchy to aggregate. ALL is the default value. Therefore, if *n* is omitted, all children are included in the sum. If *n* is 1, direct children are included. If *n* is 2, direct children and grandchildren are included. ADD 99 included children to 99 levels. ALL is a synonym for 99.

ADD

Displays the parent and *n* levels of its children on one row, summing the numeric data values displayed on the row. (This corresponds to the FML syntax *parentvalue* or CHILD1 OR CHILD2 OR ...)

To display the sum of just the children, you must display the parent row, display the summary row, and use a RECAP to subtract the parent row from the sum. For example:

```
FOR ...
parentvalue OVER
parentvalue ADD 1 OVER
RECAP CHILDSUM = R2-R1;
```

CAPTION

Indicates that the caption of the parent value will display for the total row.

Note that the AS CAPTION phrase is supported for tagged rows, including those that do not use the GET CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

'text'

Is a text string to use as the row title for the aggregate row. The CAPTION field defined in the Master File will not be used as the caption on the report output.

label

Is an explicit row label. Each generated row is labeled with the specified label text.

Example Displaying One Summary Line for an FML Hierarchy

The CENTSYSF data source contains detail level financial data. To use the account hierarchy in the CENTGL data source with this financial data, the two data sources are joined. The data in CENTSYSF is stored with natural signs, which means, in financial terms, that revenues and liabilities are stored as negative numbers. The portion of the hierarchy used in this request contains only positive data.

Note that the join is not required to be unique because the hierarchy is defined in the host segment.

First the lines of the hierarchy starting with account 3100 (Selling Expenses) are displayed by the WITH CHILDREN command. Note that only accounts with no children are populated in this detail level data source. The ADD command then creates one line that is the sum of account 3100 and all of its children:

```
SET BLANKINDENT=ON
SET FORMULTIPLE = ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
3100 WITH CHILDREN ALL AS CAPTION OVER
BAR                                OVER
3100 ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

The output is:

| | Month | YTD |
|----------------------|---------------|---------------|
| | <u>Actual</u> | <u>Actual</u> |
| Selling Expenses | . | . |
| Advertising | . | . |
| TV/Radio | 1,049,146. | 2,954,342. |
| Print Media | 244,589. | 721,448. |
| Internet Advertising | 9,542. | 29,578. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |
| | <hr/> | <hr/> |
| Selling Expenses | 1,554,319. | 4,451,098. |

Syntax

How to Consolidate FML Hierarchy Data to Any Level and Depth

```
TABLE FILE filename
SUM....
FOR hierarchyfld
parentvalue {GET|WITH} CHILD[REN] [n|ALL] ADD [m|ALL]
  [AS CAPTION 'text'] [LABEL label]
.
.
.
END
```

where:

filename

Is the name of the file to be used in the FML request. (If the hierarchy for this request cannot be loaded automatically, you must have previously issued the LOAD CHART command to load the hierarchy.)

hierarchyfld

Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

parentvalue

Is the parent value that determines the starting point in the hierarchy for the aggregation.

GET | WITH

GET specifies that the first line generated on the report is the consolidated line for the first child of the parent value. WITH specifies that the first line generated on the report is the consolidated line for the parent value, followed by the consolidated lines for each of its children to the level specified by *n*.

***n* | ALL**

Is a positive integer from 1 to 99 specifying the number of levels children to display. The line of output for each child will have the sum of that child and its children to the depth specified for the ADD option. The default value is 1. Therefore, if *n* is omitted, direct children each have a line on the report. If *n* is 2, direct children and grandchildren each have a line on the report output. ALL is a synonym for 99.

ADD

For each line generated by the GET or WITH CHILDREN command, sums the hierarchy to the depth specified by *m*.

***m* | ALL**

Is a positive integer from 1 to 99 specifying the number of levels children to consolidate on each line of the report output. If a number greater than 99 is specified, a warning message is displayed and *n* is set to 99. The default value is ALL. Therefore, if *m* is omitted, the consolidated line will sum all children. If *m* is 2, direct children and grandchildren will be consolidated for each line on the report output. ADD 99 aggregates children to 99 levels. ALL is a synonym for 99.

CAPTION

Indicates that the caption of the parent value will display for the total row.

Note that the AS CAPTION phrase is supported for tagged rows, including those that do not use the GET CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

' *text* '

Is a text string to use as the row title for the aggregate row. The CAPTION field defined in the Master File will not be used as the caption on the report output.

label

Is an explicit row label. Each generated row is labeled with the specified label text.

Example Consolidating FML Hierarchy Data

In the following request, the first WITH CHILD command displays the detail data for the hierarchy starting with account 3100. The next WITH CHILD command creates a consolidated line for the parent account (3100) and each direct child:

```

SET BLANKINDENT = ON
SET FORMULTIPLE = ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
3100 WITH CHILDREN ALL AS CAPTION          OVER
" "                                         OVER

BAR AS =                                  OVER
" "                                         OVER

3100 WITH CHILDREN  ADD AS CAPTION
IF PERIOD EQ '2002/03'
END

```

Note that the join is not required to be unique because the hierarchy is defined in the host segment.

In the following output, the top portion shows the detail level data. The bottom portion shows the consolidated data. In the consolidated portion of the report:

- There is one line for the parent that is the sum of itself plus all of its children to all levels.
- There is one line for each direct child of account 3100 (Selling Expenses): Advertising, Promotional Expenses, Joint Marketing, and Bonuses/Commissions.
- The line for Advertising is the sum of itself plus all of its children. If it had had multiple levels of children, they would all have been added into the sum. The other direct children of 3100 did not themselves have children, so the sum on each of those lines consists of only the parent value.

| | Month
<u>Actual</u> | YTD
<u>Actual</u> |
|----------------------|------------------------|----------------------|
| Selling Expenses | . | . |
| Advertising | . | . |
| TV/Radio | 1,049,146. | 2,954,342. |
| Print Media | 244,589. | 721,448. |
| Internet Advertising | 9,542. | 29,578. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |
| | <hr/> | <hr/> |
| Selling Expenses | 1,554,319. | 4,451,098. |
| Advertising | 1,303,277. | 3,705,368. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |

Using GET CHILDREN instead of WITH CHILDREN eliminates the top line from each portion of the output. The remaining lines are the same:

| | Month
<u>Actual</u> | YTD
<u>Actual</u> |
|----------------------|------------------------|----------------------|
| Advertising | . | . |
| TV/Radio | 1,049,146. | 2,954,342. |
| Print Media | 244,589. | 721,448. |
| Internet Advertising | 9,542. | 29,578. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |

| | | |
|----------------------|------------|------------|
| Advertising | 1,303,277. | 3,705,368. |
| Promotional Expenses | 53,719. | 151,732. |
| Joint Marketing | 97,135. | 289,799. |
| Bonuses/Commissions | 100,188. | 304,199. |

The following request displays a consolidated line for account 2000 and each of its direct children and grandchildren:

```

SET FORMULTIPLE = ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
2000 WITH CHILDREN 2 ADD AS CAPTION
IF PERIOD EQ '2002/03'
END

```

The output is:

| | Month
<u>Actual</u> | YTD
<u>Actual</u> |
|-------------------------|------------------------|----------------------|
| Gross Margin | -4,513,659. | -13,080,549. |
| Sales Revenue | -10,398,305. | -30,877,546. |
| Retail Sales | -8,237,253. | -24,539,197. |
| Mail Order Sales | -1,138,414. | -3,403,387. |
| Internet Sales | -1,022,638. | -2,934,962. |
| Cost Of Goods Sold | 5,884,646. | 17,796,997. |
| Variable Material Costs | 4,415,560. | 13,410,629. |
| Direct Labor | 961,143. | 2,920,449. |
| Fixed Costs | 507,943. | 1,465,919. |

Loading a Hierarchy Manually

In most cases, the hierarchy is loaded automatically as a result of the request syntax. However, if you need to use a hierarchy defined in one Master File against a data source that is not joined to the hierarchy file (but that contains the same hierarchy field), you can manually load the hierarchy data using the LOAD CHART command.

The number of charts that can be loaded is 16. Charts are automatically unloaded when the session ends.

The chart is loaded by running a TABLE request that produces a list of parent values and their associated children:

```
TABLE FILE chartfile
BY parentfield BY hierarchyfield
[SUM captionfield]
END
```

The resulting chart contains the following information. It may also contain the associated captions, depending on whether the AS CAPTION phrase was used in the request:

| <i>parentfield</i> | <i>hierarchyfield</i> |
|---------------------|-----------------------|
| ----- | ----- |
| <i>parentvalue1</i> | <i>child1</i> |
| <i>parentvalue1</i> | <i>child2</i> |
| <i>parentvalue1</i> | <i>child3</i> |
| . | |
| . | |
| . | |

Syntax**How to Load a Hierarchy From One Master File for Use With a Separate Master File**

You can manually load the hierarchy data if you need to use a hierarchy defined in one Master File against a data source that is not joined to the hierarchy file (but that contains the same hierarchy field).

The number of charts that can be loaded is limited by available memory. Charts are automatically unloaded when FOCUS terminates.

```
LOAD CHART chartfile[.sega].hierarchyfld
  [FOR requestfile[[.segb].fieldb]]
```

where:

chartfile

Is the name of the Master File that contains the hierarchy information.

sega

Is the name of the segment that contains the hierarchy field. The segment name is only required if a field in another segment in the structure has the same field name as the hierarchy field.

hierarchyfld

Is the hierarchy field. The hierarchy field is required because a Master File can define multiple hierarchies.

FOR

Loads a hierarchy defined in a Master File that will not be used in the FML report request. For example, Master File B contains the hierarchy information but Master File A will be used in the request (without a join between Master Files A and B), you would issue the following LOAD CHART command prior to the FML request:

```
LOAD CHART B.FLDB FOR A.FLDA
TABLE FILE A ...
```

requestfile

Is the name of the Master File used in the FML request.

segb

Is the name of the segment that contains the hierarchy field values in the Master File used in the FML request. Not required if it has the same name as *sega*.

fieldb

Is the field in the Master File specified in the FML request that contains the values of the hierarchy field. Not required if it has the same name as the hierarchy field.

Note:

- If you issue the LOAD CHART command multiple times for the same hierarchy, the new hierarchy overlays the previous version in memory.
- If you issue the LOAD CHART command for a data source that is dynamically joined to the hierarchy file, you must issue the JOIN command prior to issuing the LOAD CHART command.

Reference Usage Notes for FML Hierarchies

- PROPERTY and REFERENCE are propagated to HOLD Master Files when HOLDATTR is set to ON.
- The following setting is required in order to use a data record in more than one row of an FML request (for example, both a detail and summary row):

`SET FORMULTIPLE = ON`

- When reporting against a rolled up data source such as ESSBASE, the data values stored for the parent instance is an aggregate of all of its children. Do not use the ADD feature on consolidated data.
- When reporting against a data source with shared members (such as ESSBASE), in which the same data can be defined multiple times with different hierarchy field values, data shared by two different parents will be counted twice in an aggregation operation. To avoid this double aggregation, use the FST. operator in the SUM command for the shared fields.

Customizing a Row Title

You can customize a row title in an FML report for accurate data identification. Using the AS phrase, you can provide new titles for TAG, DATA, RECAP, and PICKUP rows.

Syntax **How to Customize a Row Title in FML**

For a TAG row, use the syntax

```
value AS { 'title' | CAPTION }
```

For a DATA or PICKUP row, use the syntax

```
value AS { 'title' }
```

For a RECAP row, use the syntax

```
RECAP calcname[ / format ] = expression; AS 'title'
```

where:

value

Is the value on which you are reporting, whether retrieved from a data source or external file (represented by a tag), supplied directly by a user in the request, or picked up from a work file.

calcname

Is the value that is derived by the RECAP calculation.

title

Is the customized row title, enclosed in single quotation marks if it contains embedded blanks.

In a TAG, DATA, or PICKUP row, the default row title is *value*.

In a RECAP row, the default title is *calcname*.

CAPTION

In the Master File of a hierarchical data source, CAPTION identifies a TAG row using a caption field. (Note that the hierarchy in the Master File defines the PARENT-OF the FOR field. For details, see the *Describing Data* manual.) For an illustration, see *Consolidating FML Hierarchy Data* on page A-51.

Example Changing the Titles of Tag Rows

In the following example, the row titles CASH ON HAND and DEMAND DEPOSITS provide meaningful identifications for the corresponding tags.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS'
END
```

Note that single quotation marks are necessary since the row title being assigned has embedded blanks.

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |

If no AS phrase were included, the tag values would be displayed in the report.

Example Customizing a Row Title for a RECAP Value

This request creates the title TOTAL CASH for the RECAP value TOTCASH:

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | 21,239 |

If no AS phrase were included, the name of the RECAP value (TOTCASH) would be displayed in the report.

Formatting an FML Report

You can improve the readability and presentation of your FML report by formatting reports:

- **Underlining numeric columns.** Reports with columns of numbers frequently need to display underlines before some RECAP calculations. You can specify an underline character introduced by the word BAR, in place of the tag value.
- **Adding page breaks.** You can request a new page at any point in a report by placing the word PAGE-BREAK in place of the tag value.
- **Formatting rows, columns, and cells.** You can apply StyleSheet attributes, such as FONT, SIZE, STYLE, and COLOR, to individual rows, and columns or cells within those rows.
- **Adding borders around rows, columns, and cells:** You can use BORDER attributes in a StyleSheet to specify the weight, style, and color of border lines around a row or cell. If you wish, you can specify formatting variations for the top, bottom, left, and right borders.

Syntax

How to Add an Underline Character for Columns

The syntax is

```
BAR [AS 'character' ] OVER
```

where:

character

Is either the hyphen character (-) or the equal character (=). Enclose the character in single quotation marks. The default character is the hyphen (-).

Example Underlining Columns

This example uses the default underscore character (-):

```
SET ONLINE-FMT=HTML
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'          OVER
1020 AS 'DEMAND DEPOSITS'      OVER
1030 AS 'TIME DEPOSITS'        OVER
BAR                             OVER
RECAP TOTCASH = R1 + R2 + R3;
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTCASH | <u>21,239</u> |

Notice that the BAR ... OVER phrases underline only the column containing the display field.

Syntax How to Specify a Page Break in an FML Report

To specify a page break, include the following syntax in the FML request in place of a tag value:

```
PAGE-BREAK OVER
```


Example Specifying a Page Break in an FML Report

In this example, a page break is inserted after the first two RECAP commands to highlight each calculation.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'                                OVER
1020 AS 'DEMAND DEPOSITS'                              OVER
1030 AS 'TIME DEPOSITS'                                OVER
BAR                                                    OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'          OVER
PAGE-BREAK                                           OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL RECEIVE           OVER
1200 AS 'INVENTORY' LABEL INVENT                     OVER
BAR                                                    OVER
RECAP TOTASSET = RECEIVE + INVENT; AS 'TOTAL ASSETS'  OVER
PAGE-BREAK                                           OVER
RECAP TOTAL = TOTCASH + TOTASSET;
END
```

The output is:

PAGE 1

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| <hr/> | |
| TOTAL CASH | 21,239 |

PAGE 2

| | <u>AMOUNT</u> |
|---------------------|---------------|
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |
| <hr/> | |
| TOTAL ASSETS | 46,136 |

PAGE 3

Syntax**How to Format a Row, Column, or Cell in an FML Report**

TYPE = *type*, [COLUMN=*column*] [LABEL = {*Rn*|*label*}], *format_def*, \$

where:

type

Identifies the component you wish to format. Valid values are:

REPORT

Denotes a row with the specified label.

DATA

Denotes a row with the specified label, which contains user-supplied data values.

FREETEXT

Denotes a free text or blank row with the specified label.

UNDERLINE

Denotes underlines generated by BAR.

Formatting of an underline is supported for PDF and PS, but not for HTML reports.

column

Identifies a specific column. You can identify the column by its name or its position in a row.

LABEL

Is the controlling factor in identifying and formatting an FML row.

Note that the label is used to identify a row for calculation or formatting. The label for a TAG or DATA row will never appear in the report output; it is used only to identify rows within the FML code. For a RECAP row, the name of the calculated value serves as a label; it will appear in the report unless an alternate title is specified.

The label can be implicit or explicit:

Rn

Is an implicit row label. It is assigned automatically by FML, and retained internally for processing even if you set an explicit label. To determine the value of *n*, count the number of rows up to and including the desired row.

label

Is an explicit row label that you can assign to identify a row more clearly.

format_def

Is the formatting definition, such as FONT, SIZE, STYLE, and COLOR. For details, see the *Creating Reports With WebFOCUS Language* manual.

Note: To format a cell, identify the cell as the intersection of a column and row using COLUMN=... plus LABEL= ... in the same StyleSheet declaration. For an illustration, see *Applying Boldface to a Cell in an FML Matrix* on page A-66.

Example Formatting Rows in an FML Report

The following illustrates how to identify and format an entire FML row, consisting of the row label and the row data. The LABEL attribute in the StyleSheet identifies the three TAG rows, which are styled here as italic.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL COH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, LABEL=COH, STYLE=ITALIC, $
TYPE=REPORT, LABEL=DD, STYLE=ITALIC, $
TYPE=REPORT, LABEL=TD, STYLE=ITALIC, $
ENDSTYLE
END
```

The output is:

| | <u>AMOUNT</u> |
|------------------------|---------------|
| <i>CASH ON HAND</i> | <i>8,784</i> |
| <i>DEMAND DEPOSITS</i> | <i>4,494</i> |
| <i>TIME DEPOSITS</i> | <i>7,961</i> |
| <hr/> | |
| TOTAL CASH | 21,239 |

Example Applying Boldface to a TAG Row in an FML Report

This request applies boldface to the customized row title, CASH, and to the related data in the AMOUNT column. The StyleSheet uses the explicit label CA to identify the component to format:

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                LABEL CA    OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR    OVER
1200 AS 'INVENTORY'           LABEL INV   OVER
RECAP CURASST/I5C = CA + AR + INV;
ON TABLE SET STYLE SHEET *
TYPE=REPORT, GRID=OFF, $
TYPE = REPORT, LABEL = CA, STYLE = BOLD, $
ENDSTYLE
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| CASH | 21,239 |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |
| CURASST | 67,375 |

Example Applying Boldface to a Cell in an FML Matrix

This request generates a report in which the data value for AMOUNT is bold in the row titled CASH, however, the row title CASH is not bold. This is accomplished by pinpointing the cell in the StyleSheet declaration—that is, the column (N2) within the row (CA).

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                LABEL CA    OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR      OVER
1200 AS 'INVENTORY'           LABEL INV    OVER
RECAP CURASST/I5C = CA + AR + INV;
ON TABLE SET STYLE SHEET *
TYPE=REPORT, GRID=OFF, $
TYPE = REPORT, COLUMN=N2, LABEL = CA, STYLE = BOLD, $
ENDSTYLE
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| CASH | 21,239 |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |
| CURASST | 67,375 |

Example Applying BOLDFACE to a Column in an FML Report

This request identifies the AMOUNT column by name and formats its title and data in bold. (The same result would be achieved if the column were identified as N2.)

```

SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"---CASH ACCOUNTS---"                OVER
1010 AS 'CASH ON HAND'                OVER
1020 AS 'DEMAND DEPOSITS'            OVER
1030 AS 'TIME DEPOSITS'              OVER
"  "                                  OVER
"---OTHER CURRENT ASSETS---"         OVER
1100 AS 'ACCOUNTS RECEIVABLE'        OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, COLUMN=AMOUNT, STYLE=BOLD, $
ENDSTYLE
END

```

The output is:

| | <u>AMOUNT</u> |
|----------------------------|----------------------|
| ---CASH ACCOUNTS--- | |
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| ---OTHER CURRENT ASSETS--- | |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Example Applying **BOLD**FACE to a Free Text Row

This request styles the free text as bold. Since, in this example, the same styling applies to both free text rows, labels are not required to distinguish between them.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"---CASH ACCOUNTS---" LABEL CA OVER
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
" " OVER
"---OTHER CURRENT ASSETS---" LABEL OCA OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE = FREETEXT, STYLE = BOLD, $
ENDSTYLE
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------------------------|---------------|
| ---CASH ACCOUNTS--- | |
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| ---OTHER CURRENT ASSETS--- | |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Example Formatting Free Text Rows Separately in an FML Report

This request uses the SIZE attribute to distinguish two lines of free text: CASH ACCOUNTS and OTHER CURRENT ASSETS. The labels CA and OCA are used to identify and format each row separately.

```

SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---" LABEL CA          OVER
1010 AS 'CASH ON HAND'                      OVER
1020 AS 'DEMAND DEPOSITS'                    OVER
1030 AS 'TIME DEPOSITS'                      OVER
" "                                           OVER
" --- OTHER CURRENT ASSETS ---" LABEL OCA    OVER
1100 AS 'ACCOUNTS RECEIVABLE'                OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=FREETEXT, LABEL=CA, STYLE=BOLD, SIZE=12, $
TYPE=FREETEXT, LABEL=OCA, STYLE=BOLD, SIZE=10, $
ENDSTYLE
END

```

The output is:

| | <u>AMOUNT</u> |
|-------------------------------------|---------------|
| --- CASH ACCOUNTS --- | |
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| --- OTHER CURRENT ASSETS --- | |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |

Example Styling Text and a Variable in a Free Text Row

In this example, the text and variable components of the free text row are styled separately. The text, Current Assets, is italic; the value derived from the RECAP calculation is bold.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT AS 'Amount' FOR ACCOUNT
10$$ AS 'Cash' LABEL CA OVER
1100 AS 'Accounts Receivable' LABEL AR OVER
1200 AS 'Inventory' LABEL INV OVER
RECAP CURASST/I5C= CA + AR + INV; NOPRINT OVER
"Current Assets: <CURASST"
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=FREETEXT, OBJECT=TEXT, ITEM=1, SIZE=12, STYLE=ITALIC, $
TYPE=FREETEXT, OBJECT=FIELD, ITEM=1, STYLE=BOLD, $
ENDSTYLE
END
```

The output is:

| | <u>Amount</u> |
|------------------------|---------------|
| Cash | 21,239 |
| Accounts Receivable | 18,829 |
| Inventory | 27,307 |
| <i>Current Assets:</i> | 67,375 |

Example Applying Boldface to an FML RECAP Row

This request applies boldface to the row title and calculated value in a RECAP row. Notice that the RECAP label in the StyleSheet is TOTCASH. In a RECAP, the name assigned to the calculated value serves as the explicit label.

```

SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      LABEL CASH      OVER
1020 AS 'DEMAND DEPOSITS'  LABEL DD      OVER
1030 AS 'TIME DEPOSITS'    LABEL TD      OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, LABEL= TOTCASH, STYLE= BOLD, $
TYPE=REPORT, LABEL=CASH, COLUMN=N1, STYLE=ITALIC, $
TYPE=REPORT, LABEL=DD, COLUMN=N1, STYLE=ITALIC, $
TYPE=REPORT, LABEL=TD, COLUMN=N1, STYLE=ITALIC, $
ENDSTYLE
END

```

The output is:

| | <u>AMOUNT</u> |
|------------------------|---------------|
| <i>CASH ON HAND</i> | 8,784 |
| <i>DEMAND DEPOSITS</i> | 4,494 |
| <i>TIME DEPOSITS</i> | 7,961 |
| TOTAL CASH | 21,239 |

Syntax **How to Add and Format Row and Cell Borders**

To request a uniform border around a row or cell, use this syntax

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER=option,  
[BORDER-STYLE=line_style,] [BORDER-COLOR={color|RGB(r g b)},] $
```

To specify different characteristics for the top, bottom, left, and/or right borders, use this syntax

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER-position=option,  
[BORDER-[position-]STYLE=line_style,]  
[BORDER-[position-]COLOR={color|RGB(r g b)},] $
```

where:

row_label

Is the row to which the specified border characteristics will be applied.

column

Used in conjunction with row label, designates a cell (at the point of intersection of the row and the column) to which the specified border characteristics will be applied.

option

Can be one of the following values:

ON

Turns borders on for the entire heading or footing. (ON generates the same line as MEDIUM.)

Note that the medium line setting ensures consistency with lines created with GRID attributes.

OFF

Turns borders off for the entire heading or footing. OFF is the default.

LIGHT

Specifies a thin line. You can specify a light line for the entire heading or footing or for one or more border positions.

MEDIUM

Identifies a medium line. (ON sets the line as MEDIUM.) You can specify a light line for the entire heading or footing or for one or more border positions.

HEAVY

Identifies a thick line. You can specify a heavy line for the entire heading or footing or for one or more border positions.

width

Specifies the line width in points (where 72 pts=1 inch). You can specify a line width in points for the entire heading or footing or for one or more border positions. Tip: Line width specified in points is displayed differently in HTML and PDF output. For uniform appearance, regardless of display format, use LIGHT, MEDIUM, or HEAVY.

position

Specifies which border line to format. Valid values are: TOP, BOTTOM, LEFT, RIGHT.

You can specify a position qualifier for any of the BORDER keywords This enables you to format line width, line style, and line color individually, for any side of the border.

line style

Sets the style of the border line. options. WebFOCUS StyleSheets support all of the standard Cascading Style Sheets line styles. Several 3-dimensional styles are only available in HTML, as noted by asterisks. Valid values are:

NONE
SOLID
DOTTED
DASHED
DOUBLE
GROOVE*
RIDGE*
INSET*
OUTSET*

color

Is one of the preset color values. The default value is BLACK.

If the display or output device does not support colors, it substitutes shades of gray. For a complete list of available color values, see Chapter *the Creating Reports With WebFOCUS Language* manual.

RGB

Specifies the font color using a mixture of red, green, and blue.

(r g b)

Is the desired intensity of red, green, and blue, respectively. The values are on a scale of 0 to 255, where 0 is the least intense and 255 is the most intense. Note that using the three color components in equal intensities results in shades of gray.

Note: For HTML reports, the BORDERS feature requires that Cascading Style Sheets be turned ON. This code is not required for PDF and PS reports.

Example Emphasizing a Row Using Uniform Border Lines

This example places a thick dashed border around the RECAP row identified by the label TOTCASH. For HTML reports, the BORDERS feature requires that Cascading Style Sheets be turned ON.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      LABEL CASH      OVER
1020 AS 'DEMAND DEPOSITS'  LABEL DD        OVER
1030 AS 'TIME DEPOSITS'    LABEL TD        OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, LABEL=TOTCASH, BORDER=MEDIUM,
    BORDER-STYLE=DASHED, $
ENDSTYLE
END
```

The output is:

| | AMOUNT |
|-----------------|--------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | 21,239 |

Example Emphasizing a Row Using Different Top/Bottom and Left/Right Borders

This example places a heavy black border line above and below the the RECAP row identified by the label TOTCASH, and a thin silver dotted line to the left and right of each column in the row.

For HTML reports, the BORDERS feature requires that Cascading Style Sheets be turned ON.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      LABEL CASH      OVER
1020 AS 'DEMAND DEPOSITS'  LABEL DD        OVER
1030 AS 'TIME DEPOSITS'    LABEL TD        OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, LABEL=TOTCASH,
    BORDER-TOP=HEAVY,
    BORDER-BOTTOM=HEAVY,
    BORDER-LEFT=LIGHT,
    BORDER-RIGHT=LIGHT,
    BORDER-TOP-STYLE=SOLID,
    BORDER-BOTTOM-STYLE=SOLID,
    BORDER-LEFT-STYLE=DOTTED,
    BORDER-RIGHT-STYLE=DOTTED,
    BORDER-LEFT-COLOR='SILVER',
    BORDER-RIGHT-COLOR='SILVER', $
ENDSTYLE
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | 21,239 |

Example Adding Uniform Border Lines Around a Cell

This example places a thick black border around the cell in the second column of the row identified by the label TOTCASH. The combined LABEL and COLUMN specifications identify in the cell. The BORDERS feature requires that Cascading Style Sheets be turned ON.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      LABEL CASH      OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD        OVER
1030 AS 'TIME DEPOSITS'    LABEL TD        OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
TYPE=REPORT, LABEL=TOTCASH, COLUMN=N2, BORDER=MEDIUM, $
ENDSTYLE
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| CASH ON HAND | 8,784 |
| DEMAND DEPOSITS | 4,494 |
| TIME DEPOSITS | 7,961 |
| TOTAL CASH | 21,239 |

Suppressing the Display of Rows

You may sometimes wish to retrieve data in a TAG row solely for use in a calculation, without displaying the row in a report. To suppress the display of a tag row, you can add the word NOPRINT to the row declaration, as you would in a TABLE request.

You may also wish to suppress the display of a TAG row if no data is found for the values. For details see *Suppressing Rows With No Data* on page A-78.

In addition, you can suppress the display of RECAP rows by adding the word NOPRINT to the RECAP command, following the semicolon (see *Styling Text and a Variable in a Free Text Row* on page A-70). This technique is particularly useful to suppress the display of an intermediate RECAP value, which is intended for use as input to other calculations.

Example Suppressing the Display of a TAG Row

This example uses the value of COST in its computation, but does not display COST as a row in the report.

```
DEFINE FILE REGION
AMOUNT/I5C=E_ACTUAL;
END

TABLE FILE REGION
SUM AMOUNT FOR ACCOUNT
3000 AS 'SALES' LABEL SLS OVER
3100 AS 'COST' LABEL COST NOPRINT OVER
RECAP PROFIT/I5C = SLS - COST; OVER
" " OVER
RECAP ROS/F6.2=100*PROFIT/SLS;
AS 'RETURN ON SALES'
END
```

The output is:

| | <u>AMOUNT</u> |
|-----------------|---------------|
| SALES | 6,000 |
| PROFIT | 1,350 |
| RETURN ON SALES | 22.50 |

Suppressing Rows With No Data

The text for a tag row is displayed even if no data is found in the file for the tag values, with a period (.) representing the missing data. You can override this convention by adding the phrase WHEN EXISTS to the definition of a TAG row. This makes displaying a row dependent upon the existence of data for the tag. This feature is particularly useful, for example, when the same model is applied to different divisions in a company.

Example Suppressing Rows With No Data

In this example, assume that the variable DIVISION contains Division 1, a real estate syndicate, and Division 2, a bank. The following request describes their balance sheets in one FML report. Rows that are irrelevant for each division will not be displayed.

```
TABLE FILE LEDGER
HEADING CENTER
"BALANCE SHEET FOR DIVISION <DIVISION>"
"  "
SUM AMOUNT
BY DIVISION NOPRINT
ON DIVISION PAGE-BREAK
FOR ACCOUNT
2000 AS 'LAND' WHEN EXISTS LABEL LD          OVER
2100 AS 'CAR LOANS' WHEN EXISTS LABEL LOAN    OVER
.
.
.
```

Saving and Retrieving Intermediate Report Results

Many reports require results developed in prior reports. This can be accomplished only if a place is provided for storing intermediate values. An example is the need to compute net profit in an Income Statement prior to calculating equity in a Balance Sheet. FML can save selected rows from one or more models by posting them to a work file. The posted rows can then be picked up from the work file and reused.

The default work file is FOCPOST. This is a comma-delimited file from which you can report directly if a FOCPOST Master File is available. In order to use the work file in a request, you must assign a physical name to the FOCPOST ddname before running the report that posts the data, and again before running the report that picks up the data.

You can assign the physical name to the file by issuing a FILEDEF command on NT, UNIX, and CMS, or a TSO ALLOCATE or DYNAM ALLOCATE command on MVS, before the request is run.

While you cannot prepare an FML report entirely from data that you supply directly in your request, you can, if you wish, prepare a report entirely from data that is stored in a comma-delimited work file.

Posting Data

You can save any TAG, RECAP, or DATA row by posting the output to a file. These rows can then be used as though they were provided in a DATA row.

The row will be processed in the usual manner in the report, depending on its other options, and then posted. The label of the row is written first, followed by the numeric values of the columns, each comma-separated, and ending with the terminator character (\$). For an illustration see *Posting Rows to a Work File* on page A-80.

Syntax

How to Post Data to a File

The syntax for saving any TAG, RECAP, or DATA row is:

```
POST [TO ddname]
```

where:

ddname

Is the logical name you assign to the work file in which you are posting data.

You add this syntax to any row you wish to post to the work file.

Example Posting Rows to a Work File

The following request create an FML report, and posts two tag rows to the work file, LEDGEOUT:

```
FILEDEF LEDGEOUT DISK [PATH]\LEDGEOUT.DAT
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR LAST_YR
FOR ACCOUNT
1100 LABEL AR POST TO LEDGEOUT OVER
1200 LABEL INV POST TO LEDGEOUT
END
```

The output is:

| | <u>CUR_YR</u> | <u>LAST_YR</u> |
|------|---------------|----------------|
| 1100 | 18,829 | 15,954 |
| 1200 | 27,307 | 23,329 |

Syntax How to Pick Up Data From a Work File

You can retrieve posted rows from any work file and use them as if they were provided in a DATA row by adding the following phrase to an FML request

```
DATA PICKUP [FROM ddname] id1 [OR id2....] [LABEL label1] [AS 'text']
```

where:

ddname

Is the logical name of the work file from which you are retrieving data.

id

Is the label that was assigned in the work file to the posted row of data that is now being picked up.

label1

Is the label you wish to assign to the data you are picking up.

The label you assign to the picked data can, but is not required to, match the id of the posted data.

You can include LABEL and AS phrases, but WHEN EXISTS is not supported.

Example Picking Up Data From a Work File

In the following example, the data in the LEDGER data source and in the LEDGEOUT work file will be used in the RECAP calculation. (To see how this file was created refer to *Posting Rows to a Work File* on page A-80.)

Tip: You must assign a logical name to the file by issuing a FILEDEF command on NT, UNIX, and CMS, or a DYNAM ALLOCATE command on MVS, before the request is run.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR LAST_YR
FOR ACCOUNT
1010 TO 1030 AS 'CASH' LABEL CASH          OVER
DATA PICKUP FROM LEDGEOUT AR
AS 'ACCOUNTS RECEIVABLE' LABEL AR          OVER
DATA PICKUP FROM LEDGEOUT INV
AS 'INVENTORY' LABEL INV                  OVER
BAR                                         OVER

RECAP CUR_ASSET/I5C = CASH + AR + INV;
END
```

The output is:

| | <u>CUR_YR</u> | <u>LAST_YR</u> |
|---------------------|---------------|----------------|
| CASH | 21,239 | 17,198 |
| ACCOUNTS RECEIVABLE | 18,829 | 15,954 |
| INVENTORY | 27,307 | 23,329 |
| CUR_ASSET | <u>67,375</u> | <u>56,481</u> |

The following line could be used to pick up the sum of the two accounts from LEDGEOUT:

```
DATA PICKUP FROM LEDGEOUT AR OR INV
AS 'ACCTS REC AND INVENTORY'
```

Note: Since the rows in a PICKUP file are stored in standard comma-delimited format, they could have been provided either from a prior posting, or directly by a user.

Creating HOLD Files From FML Reports

A report created with FML can be extracted to a HOLD file in the same way as all other reports created with the TABLE language (see Chapter 9, *Saving and Reusing Your Report Output* in the *Creating Reports With WebFOCUS Language* manual). In this case, you identify the set of tag values specified for each row by the description field (the AS text supplied in the model). When no text is given for a row, the first tag value is used automatically. Therefore, in simple models with only one tag per row and no text, the lines in the HOLD file contain the single tag value. The rows derived from the RECAP calculation form part of the HOLD file. Pure text rows (including BAR rows) are omitted.

For HOLD to be supported with RECAP, the format of the RECAP field must be the same as the format of the original column.

This feature enables you to create new rows in the HOLD file that are the result of calculations. The augmented HOLD file may then be used in a variety of TABLE requests.

Note:RECAP rows cannot be reformatted when creating HOLD files.

Example Creating a Hold File From an FML Report

The following request creates a HOLD file that contains records for CASH, ACCOUNTS RECEIVABLE, INVENTORY, and the RECAP row CURRENT ASSETS:

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 TO 1030 AS 'CASH'                                OVER
1100 AS 'ACCOUNTS RECEIVABLE'                        OVER
1200 AS 'INVENTORY'                                  OVER
RECAP CA = R1 + R2 + R3; AS 'CURRENT ASSETS'
ON TABLE HOLD
END
```

You can query the HOLD file:

```
>
? hold
```

```
DEFINITION OF HOLD FILE: HOLD
```

| FIELDNAME | ALIAS | FORMAT |
|-----------|-------|--------|
| | E01 | A19 |
| AMOUNT | E02 | I5C |

You can then report from the HOLD file as:

```
TABLE FILE HOLD
PRINT E01 E02
END
```

The output is:

| | <u>AMOUNT</u> |
|---------------------|---------------|
| CASH | 21,239 |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY | 27,307 |
| CURRENT ASSETS | 67,375 |

APPENDIX B

Describing Data for an FML Hierarchy

Topics:

- Data Requirements for FML Hierarchies
- Coding an FML Hierarchy in a Text Editor
- Defining an FML Hierarchy in the Master File Editor

This manual focuses on WebFOCUS financial reporting with the Financial Reporting Painter, supported by the underlying Financial Modeling Language (FML).

It includes detailed information about reporting against hierarchical data structures. For your convenience, this appendix provides the information you need to understand how these hierarchical relationships between fields are described in a Master File.

As with other Master File attributes, you can define the hierarchy using either a text editor or the graphical Master File Editor.

For detailed information about all aspects of the data description language, see the *Describing Data With WebFOCUS Language* and/or the *Describing Data With Graphical Tools* manuals.

Data Requirements for FML Hierarchies

In the Master File, use the `PROPERTY=PARENT_OF` and `REFERENCE=hierarchyfld` attributes to define the hierarchical relationship between two fields.

To use FML hierarchies, the FOR field must either be:

- The hierarchy field.
- or
- Used as the join field to a unique segment that has the hierarchy field. In this case the hierarchy field must be the join field. Note that the condition that the join be unique only applies if the hierarchy is defined in the cross-referenced segment.

In other words, the FOR field must be in a parent-child hierarchy or it must be linked to a parent-child hierarchy. The latter case allows transaction data that contains the hierarchy field to be joined to a separate data source that contains the hierarchy definition.

For related information, see *Requirements for FML Hierarchies* in Appendix A, *Creating Financial Reports With FML Language*.

Coding an FML Hierarchy in a Text Editor

You can manually code the relationship between parent and child fields in a Master File, and, optionally, provide descriptive captions to display in reports in place of the specified hierarchy field values.

Syntax

How to Specify a Hierarchy Between Fields in a Master File

`FIELD=parentfield, ..., PROPERTY=PARENT_OF, REFERENCE=[seg.] hierarchyfld, $`

where:

parentfield

Is the parent field in the hierarchy.

`PROPERTY=PARENT_OF`

Identifies this field as the parent of the referenced field in a hierarchy.

These attributes can be specified on every field. Therefore, multiple hierarchies can be defined in one Master File. However, an individual field can have only one parent. If multiple fields have `PARENT_OF` attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order is used as the parent.

seg

Is the segment location of the hierarchy field. Required if more than one segment has a field named *hierarchyfield*.

hierarchyfld

Is the child field in the hierarchy.

PARENT_OF is also allowed on a virtual field in the Master File:

```
DEFINE name/ fmt=expression; , PROPERTY=PARENT_OF, REFERENCE=hierarchyfld , $
```

Syntax**How to Assign Descriptive Captions for Hierarchy Field Values**

The following attributes specify a caption for a hierarchy field in a Master File

```
FIELD=captionfield, ..., PROPERTY=CAPTION, REFERENCE=[seg.] hierarchyfld, $
```

where:

captionfield

Is the name of the field that contains the descriptive text for the hierarchy field. For example, if the employee ID is the hierarchy field, the last name may be the descriptive text to be displayed on the report in place of the ID.

PROPERTY=CAPTION

Signifies that this field contains a descriptive caption to be displayed in place of the hierarchy field values.

A caption can be specified for every field, but an individual field can have only one caption. If multiple fields have CAPTION attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order will be used as the caption.

seg

Is the segment location of the caption field. Required if more than one segment has a field named *captionfield*.

hierarchyfld

Is the hierarchy field.

CAPTION is also allowed on a virtual field in the Master File:

```
DEFINE name/ format=expression; , PROPERTY=CAPTION, REFERENCE=hierarchyfld , $
```

Example Defining a Hierarchy in a Master File

The CENTGL Master File contains a chart of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field:

```
FILE=CENTGL          , SUFFIX=FOC
SEGNAME=ACCOUNTS, SEGTYPE=S01
FIELDNAME=GL_ACCOUNT,          ALIAS=GLACCT,  FORMAT=A7,
                                TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT,    ALIAS=GLPAR,   FORMAT=A7,
                                TITLE=Parent,
                                PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE,      ALIAS=GLTYPE,  FORMAT=A1,
                                TITLE=Type, $
FIELDNAME=GL_ROLLUP_OP,         ALIAS=GLROLL,  FORMAT=A1,
                                TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL,     ALIAS=GLLEVEL, FORMAT=I3,
                                TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION,   ALIAS=GLCAP,   FORMAT=A30,
                                TITLE=Caption,
                                PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT,          ALIAS=ALINE,   FORMAT=A6,
                                TITLE='System,Account,Line', MISSING=ON, $
```

The CENTSYSF data source contains detail-level financial data. This is unconsolidated financial data for a fictional corporation, CenturyCorp. It is designed to be separate from the CENTGL database as if it came from an external accounting system. It uses a different account line system (SYS_ACCOUNT) which can be joined to the SYS_ACCOUNT field in CENTGL. Data uses “natural” signs (expenses are positive, revenue negative).

```
FILE=CENTSYSF        , SUFFIX=FOC
SEGNAME=RAWDATA      , SEGTYPE=S2
FIELDNAME = SYS_ACCOUNT , , A6 , FIELDT
PE=I,
                                TITLE='System,Account,Line', $
FIELDNAME = PERIOD      , , YYM , FIELDTYPE=I, $
FIELDNAME = NAT_AMOUNT  , , D10.0 , TITLE='Month,Actual', $
FIELDNAME = NAT_BUDGET  , , D10.0 , TITLE='Month,Budget', $
FIELDNAME = NAT_YTDAMT  , , D12.0 , TITLE='YTD,Actual', $
FIELDNAME= NAT_YTDBUD   , , D12.0 , TITLE='YTD,Budget', $
```

Defining an FML Hierarchy in the Master File Editor

As an alternative to manually coding the hierarchy attributes in a text editor, you can define them using the graphical Master File Editor. The controlling fields are: Field Property and Field Reference on the General Tab that is used to define fields in a segment.

Reference **Master File Tab: General (Field Level)**

The screenshot shows the 'General' tab of the Master File Editor. The 'Field Name' is 'GL_ACCOUNT' and the 'Index (Standard B-tree)' checkbox is checked. The 'Field Alias' is 'GLACCT'. The 'Field Property' is set to 'PARENT_OF' and the 'Field Reference' is '<None>'. The 'Format' section shows 'Alphanumeric' selected with a length of '7'. Other format options include Floating Point, Integer, Decimal, Packed, Date/Time, Text, CLOB, and VarChar. The 'Allow Missing Data' checkbox is unchecked.

| Field Name | Field Alias | Field Property | Field Reference | Format | Length | Allow Missing Data |
|------------|-------------|----------------|-----------------|--------------|--------|--------------------------|
| GL_ACCOUNT | GLACCT | PARENT_OF | <None> | Alphanumeric | 7 | <input type="checkbox"/> |

When a field is selected in the left (Master File) frame, the General tab has the following fields/options:

Field Name

Enter a unique name (1-66 characters) in uppercase characters. The first character must be a letter. Do not use embedded blanks or special characters if you wish to include the field in a calculation. This is a required entry.

Field Alias

Enter a brief alternate name that you can use in requests as a synonym for the field name to minimize typing. (Length and format rules apply to aliases.) This is an optional entry.

If you create a report, the field name appears as a column heading unless you have specified an alternate title for the field. Aliases cannot be used as column titles.

Note: Aliases are not available for virtual fields and are not used in the Maintain language.

Field Property

Choose one of the following, to be used in conjunction with Field Reference:

PARENT_OF

Identifies this field as the parent of the referenced field in a hierarchy. This attribute can be specified on every field. Therefore, multiple hierarchies can be defined in one Master File. However, an individual field can have only one parent. If multiple fields have PARENT_OF attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order is used as the parent.

CAPTION

Signifies that this field contains a descriptive caption that can be displayed in place of the hierarchy field values in a report. For example, if the employee ID is the hierarchy field, the last name may be the descriptive text to be displayed in place of the ID.

A caption can be specified for every field, but an individual field can have only one caption. If multiple fields have CAPTION attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order will be used as the caption.

You can assign either PARENT_OF or CAPTION to a field in the data source or to a virtual field defined in the Master File.

Field Reference

If PARENT is the field property, the reference is the child field in the hierarchy. This entry indicates which field is the child of the specified parent.

If CAPTION is the field property, the reference is the child field in the hierarchy. This entry indicates which field to apply the caption to.

Index (Standard B-tree)

You can index the values of a field to enhance data retrieval performance. To do so, select the Index check box when you add a field and *before* you add the data. An index is an internally stored and maintained table of data values and locations that enhance the performance of data retrieval. A Master File can have several associated indexes, but the combined total of indices and segments cannot exceed 64.

This feature is not available for virtual fields.

Tip: You can turn on the index *after* adding data to a field, however, you will have to use the Rebuild Index option to create the index.

Format

The field Format option is a required entry. It enables you to assign a format to the field based on the values the field will hold. The options are: Alphanumeric, Numeric (Floating Point, Integer, Decimal, Packed), and Date.

The input fields in the window adjust based on the format you select:

- For fields with Numeric formats, you can assign display options that determine how the field will be displayed in reports.
- For fields with a Date format, you assign display options that determine how the field will be displayed in reports. You can also assign century values to the date field.

Maintain users should see the WebFOCUS Maintain documentation for detailed information about available formats (data types) and associated display options.

Tip: The term Format in the Master File Editor is equivalent to the term Type in the WebFOCUS Maintain Type wizard.

Length (1-256)

Indicate the maximum number of positions an alphanumeric or numeric field can contain.

Allow Missing Data (do not substitute blanks or zeros)

If a segment instance exists but no data has been entered into one of its fields, that field has no value. This absence of data is represented as a null value or missing data. You use the Allow Missing Data check box to request the display of a missing data value for a field when read from a data source or written to a data source.

Not supported for virtual fields.

Index

A

accounting data 1-3
ACROSS phrase A-27
adding columns to financial reports 2-87, A-35
adding data to financial reports 2-13, 2-45, A-5, A-16
adding page breaks A-60
adding rows to financial reports 2-85, A-5
adding tags to financial reports 2-13 to 2-14, 2-16, A-6, A-42
adding text to financial reports 2-57, A-32
adding underlines to financial reports 2-64, A-59
aggregating financial data 1-5
aliases of fields B-5
archiving financial reports 1-8
assigning row labels 2-46, 2-50, 2-62, A-19
assigning row titles 2-5, A-57
 DATA rows 2-46
 PICKUP rows 2-62, A-81
 RECAP rows 2-50, A-58
 TAG rows 2-18, A-58

B

BAR command A-59 to A-60
BAR dialog box 2-64 to 2-65
borders 2-67, 2-72, 2-78, A-72
 adding around cells A-72
 adding for cells A-76
 adding for rows A-74 to A-75
budgets 1-1

bursting reports 1-7

By fields 2-50

BY phrase 2-2, A-15

C

calculations 2-47 to 2-48, 2-50, A-30
 functions and A-30 to A-31
captions
 displaying for hierarchies 2-27, A-45
 indenting 2-26 to 2-27, A-45
cell notation A-29
cells 2-66, A-28
 adding borders 2-78, A-72, A-76
 formatting 2-64, 2-72 to 2-73, 2-76, 2-78 to 2-79, 2-82, A-66, A-72
 identifying 2-66, A-28 to A-29
changing row titles 2-5, A-57
 DATA rows 2-46
 PICKUP rows 2-5, 2-18, 2-50, 2-62, A-81
 RECAP rows 2-50, A-58
 TAG rows 2-18, A-58
Charts of Accounts 1-5
charts of accounts hierarchies 2-44, A-39, A-41, B-4
charts of accounts reports 1-8, 2-21, A-39
column numbers A-23
columns 2-4 to 2-5, A-5
 adding to financial reports 2-87, A-35
 addressing A-25
 deleting from financial reports 2-87
 formatting 2-64, 2-67, 2-69, 2-72 to 2-73, 2-76, A-63, A-67
 formatting in the Design matrix 2-4
 identifying A-22 to A-26
 values for A-27

Index

- combining records A-10
- combining values 2-13 to 2-14, 2-16 to 2-18, A-8, A-10
- conditional formatting 2-83
- Conditional Styling dialog box 2-83
- consolidating financial data 1-5, 2-18, 2-23, 2-31, A-42, A-46
 - multiple rows 2-39, A-49
 - single row 2-32, A-47
 - singly row A-48
- constants 2-45, A-16 to A-17
- contiguous columns A-24
- creating financial reports 1-5
- creating HOLD files A-82 to A-83
- creating Recap expressions with Financial Report Painter 2-47
- creating rows A-5
 - from multiple records A-8
- creating tag rows 2-13, A-7

D

- data 2-45
 - adding to financial reports 2-45, A-16
- DATA dialog box 2-45
- data posting 2-60 to 2-61, A-79
- data retrieval 2-61, A-5, A-80 to A-81
- data types B-5
- defining output format with Financial Report Painter 2-47, 2-53
- deleting columns from financial reports 2-87
- deleting rows from financial reports 2-85
- delivering financial data 1-2

- describing fields B-5
- Design mode 2-10
- Design/Preview icon 2-10 to 2-11
- designing financial reports 2-4
- Dialogue Manager A-2
- display formats for fields B-5
- displaying children 2-23, 2-26, A-42, A-44 to A-45
- displaying FML hierarchies 2-26, A-39, A-42
 - requirements A-40, B-2
- displaying parents and children 2-23, 2-26, A-42, A-44 to A-45
- drill down capability 2-83

E

- Edit Condition dialog box 2-82
- editing rows with Financial Report Painter 2-86
- ESSBASE hierarchies 2-21, A-39
- event-driven report distribution 1-7
- Excel Pivot Tables 1-6
- explicit labels 2-5, 2-18, 2-46, 2-50, 2-62, A-19 to A-21
- external files 2-60, A-14

F

- field aliases B-5
- field formats B-5
- field names B-5
- field properties B-5
- Field Properties Drill Down tab 2-83
- field references B-5

- fields 2-2
 - describing B-5
 - indexing B-5
 - naming B-5
- financial consolidation 1-1
- financial data 1-3
 - consolidating 1-5, 2-31 to 2-32, 2-39, A-46 to A-49
 - delivering 1-2
 - displaying as Total Revenue 1-5
- Financial Modeling Language (FML) 1-11, 2-1, A-1 to A-3
 - Dialogue Manager and A-2
- Financial Report Painter 1-1, 1-8, 1-11, 2-1 to 2-2, 2-4 to 2-5
 - adding tags 2-13 to 2-14
 - adding text 2-57
 - adding underlines 2-64
 - advantages for data access 1-11
 - advantages over SQL tools 1-10
 - creating Recap expressions 2-47 to 2-48
 - defining output format 2-47
 - Design mode 2-10
 - editing rows 2-86
 - Preview mode 2-10 to 2-11
 - retrieving posted data 2-61, 2-63
 - saving intermediate results 2-60 to 2-61
 - suppressing row display 2-59
 - viewing source code 2-9 to 2-10
- Financial Report Painter toolbar 2-4 to 2-5
- financial reporting 1-2 to 1-3
 - government regulations for 1-4
- financial reports 1-1 to 1-2, 2-1 to 2-2, A-1 to A-2, A-39
 - adding columns to 2-87, A-35
 - analytic 1-6
 - archiving 1-8
 - charts of accounts 1-8, 2-21, 2-23, A-39, A-42, A-44 to A-45
 - consolidating data 1-5
 - constants in 2-45, A-16
 - contiguous columns in A-24
 - creating 1-5, 1-8, A-3, A-8
 - creating rows in A-5
 - designing 2-4
 - distributing 1-7
 - external files and A-14
 - for external use 1-3
 - for internal use 1-3
 - formatting 2-66, A-59 to A-60
 - hierarchies and 2-21, A-42, A-47, A-56
 - HOLD files and A-82 to A-83
 - identifying columns in A-22 to A-23
 - identifying rows in A-20
 - inserting text rows 2-57, A-32
 - inserting variables in text rows A-33
 - international currencies 1-5
 - inter-row calculations and 2-47, A-18 to A-19
 - Master Files for A-41, B-4
 - Master Files for charts of accounts 2-44
 - posting data in 2-60, A-79
 - recursive models and A-37
 - retrieving posted data 2-61, 2-63
 - running 2-12
 - saving intermediate results 2-60 to 2-61, A-79
 - sorting 2-2
 - sorting with BY A-15
 - sorting with FOR A-15
 - suppressing rows in 2-59, A-77 to A-78
 - types 1-8
- FML (Financial Modeling Language) 2-1, A-1 to A-3
 - Dialogue Manager and A-2
- FML hierarchies 2-21, 2-23, 2-44, A-39, A-41 to A-42, A-44 to A-45, B-4
 - displaying A-42
 - Master Files for 2-44, A-41, B-4

Index

FML Painter 2-1 to 2-2, 2-4 to 2-5

- adding tags 2-13 to 2-14
- adding text 2-57
- adding underlines 2-64
- creating Recap expressions 2-47 to 2-48
- defining output format 2-47, 2-53
- Design mode 2-10
- editing rows 2-86
- Preview mode 2-10 to 2-11
- retrieving posted data 2-61, 2-63
- saving intermediate results 2-60 to 2-61
- suppressing row display 2-59
- viewing source code 2-9 to 2-10

FML Painter toolbar 2-4 to 2-5

FOCPOST files 2-60 to 2-61, A-79

FOR field 2-2 to 2-5, 2-13, A-3

- reusing values 2-5

FOR phrase A-2 to A-3, A-15

- reusing values A-12

forecasting trends 1-6

Format dialog box 2-47, 2-53

formatting cells 2-78 to 2-79, 2-82, A-59, A-63, A-66

formatting columns 2-67, 2-69, A-59, A-63, A-67

formatting financial reports 2-4, 2-13, 2-18, 2-45 to 2-46, 2-59, 2-62, 2-64

formatting free text 2-76, A-69

formatting Recap rows 2-47, 2-50

formatting reports A-59 to A-60, A-63

- with boldface A-65
- with border lines A-74 to A-75

formatting rows 2-72 to 2-73, 2-76, A-59, A-63 to A-65

formatting text rows 2-76, A-68

Formula Calculator 2-47

FORMULTIPLE parameter A-12

free text A-32, A-68 to A-70

- formatting 2-76, A-69
- identifying in WebFOCUS StyleSheets A-69 to A-70

FREETEXT component A-69 to A-70

functions A-30

- calling A-30 to A-31

G

General tab in Master File Editor B-5

GET CHILDREN parameter A-42, A-47

government mandated financial reports 1-3

government regulations for financial reporting 1-4

groups of values 2-14, 2-16 to 2-17, A-11

H

hiding rows 2-59, A-77 to A-78

hierarchies A-47, A-55 to A-56

- displaying 2-26, A-39
- displaying values as captions 2-23, 2-26 to 2-27, A-45
- indenting levels 2-26

HOLD files A-82 to A-83

I

identifying cells A-28 to A-29

identifying columns A-22, A-26

- by address A-25
- by number A-23
- by relative address A-26

identifying contiguous columns A-24

identifying rows A-19 to A-21

indenting hierarchy levels 2-26

indexed fields B-5
 inserting rows in financial reports 2-85
 inserting text in financial reports 2-57, A-32
 inserting variables in free text A-33
 international currencies in reports 1-5
 inter-row calculations 2-47, A-18

K

Knowledge Mapping 1-6

L

LABEL attribute A-63 to A-65
 labels A-19
 default 2-5
 explicit 2-5, 2-18, 2-46, 2-50, 2-62, A-19
 for rows A-20 to A-21
 formatting rows and A-63 to A-65, A-69
 positional A-19

M

masking characters 2-17 to 2-18, A-11
 Master File Editor B-5
 Master Files A-40 to A-41, B-2, B-4
 hierarchies in A-40, A-55, B-2
 Master Files for hierarchies 2-44, A-41
 Matrix button 2-4
 multiple values 2-13, A-8
 combining 2-14, A-8
 identifying ranges of 2-16, A-10

N

naming fields B-5
 non-recursive models A-37
 NOPRINT command A-77
 null values B-5

O

OLAP analysis 1-6
 OR operator A-10

P

PAGE BREAK command A-60
 page breaks A-60
 PICKUP dialog box 2-60, 2-62
 positional labels A-19 to A-20
 POST command A-79
 posting data 2-60, A-79 to A-80
 posting intermediate results to a file 2-60 to 2-61
 Preview mode 2-10 to 2-11
 Properties bar 2-4 to 2-5

R

range of values 2-16, A-11
 ranges 2-13, 2-16
 RECAP command A-2 to A-3, A-18 to A-19
 RECAP dialog box 2-47, 2-50
 RECAP expression 2-48, A-19, A-21, A-25
 Recap operations 2-47 to 2-48, 2-55
 RECAP rows A-71
 formatting A-71
 identifying in WebFOCUS StyleSheets A-71
 summarizing 2-56

Index

- records A-8
 - combining A-8
 - in multiple rows A-12
 - reusing A-12
 - recursive models A-37
 - recursive reports 1-8
 - relative column addresses A-26
 - report bursting 1-7
 - report distribution 1-7
 - report fields 2-4 to 2-5
 - ReportCaster 1-7
 - reporting against hierarchies 2-21, 2-23, A-39, A-42, A-44 to A-45, A-47, A-56
 - reporting solution 1-1
 - resuing tag values 2-5
 - retrieving data values for 2-13, A-7
 - retrieving posted data 2-60 to 2-61, 2-63, A-80 to A-81
 - reusing FOR field values 2-5, A-12
 - row labels 2-4 to 2-5, A-21
 - assigning to DATA rows 2-46
 - assigning to PICKUP rows 2-62
 - assigning to RECAP rows 2-50
 - assigning to TAG rows 2-46, 2-50, 2-62
 - row titles 2-4 to 2-5, A-57 to A-58
 - row types 2-4 to 2-5
 - row-based reports 1-8
 - rows 2-85, A-5, A-7 to A-8
 - adding to financial reports 2-85
 - assigning titles to A-57 to A-58
 - calculations and A-18 to A-19
 - creating A-5, A-7 to A-8
 - deleting from financial reports 2-85
 - editing 2-86
 - formatting 2-64, 2-72 to 2-73, 2-76
 - formatting in the Design matrix 2-4
 - identifying A-19 to A-21
 - identifying in WebFOCUS StyleSheets A-63 to A-65
 - labels for A-20
 - saving A-79 to A-80
 - suppressing A-77 to A-78
 - suppressing display in financial reports 2-59
 - running financial reports 2-12
- ## S
-
- saving intermediate report results 2-60 to 2-61, A-79
 - saving intermediate results A-79
 - saving rows A-79 to A-80
 - scheduling report distribution 1-7
 - SET FORMULTIPLE parameter A-12
 - sort fields A-5, A-15
 - sort sequence A-15
 - source code 2-9
 - viewing FML code 2-9 to 2-10
 - specifying ranges in financial reports 2-13, 2-16, A-11
 - style sheets A-59
 - WebFOCUS StyleSheets A-59
 - summary RECAP rows 2-56
 - supplying data directly in FML 2-45, A-16 to A-17
 - suppressing rows 2-59, A-77 to A-78

T

TAG dialog box 2-13, 2-18

TAG rows A-77 to A-78
 suppressing A-77 to A-78

tag values 2-13
 reusing 2-5, A-6, A-12 to A-13, A-46

TEXT dialog box 2-57 to 2-58

text rows 2-57, A-32 to A-33
 formatting 2-76, A-68 to A-70

TO phrase A-11

Total Revenue 1-5

V

values 2-13, A-7 to A-8
 combining 2-14, 2-16 to 2-18, A-8, A-10
 identifying groups of 2-14, A-11
 identifying ranges of 2-16, A-10 to A-11
 in multiple rows A-12
 retrieving with masking characters 2-17, A-11
 reusing 2-5, A-12

View Focexec icon 2-9 to 2-10

W

WebFOCUS Financial Reporting Platform 1-5

WebFOCUS StyleSheets A-59

WHEN EXISTS phrase 2-59, A-78

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services - Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: (212) 967-0460

E-mail: books_info@ibi.com

Web form: <http://www.informationbuilders.com/bookstore/derf.html>

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

E-mail: _____

Comments:

Reader Comments